

Unitary-circuit semantics for measurement-based computations

Niel de Beaudrap*

Quantum Information Theory Group
Institut für Physik und Astronomie
Universität Potsdam

Abstract

One-way measurement based quantum computations (1WQC) may describe unitary transformations, via a composition of CPTP maps which are not all unitary themselves. This motivates the following decision problems: Is it possible to determine whether a “quantum-to-quantum” 1WQC procedure (having non-trivial input and output subsystems) performs a unitary transformation? Is it possible to describe precisely *how* such computations transform quantum states, by translation to a quantum circuit of comparable complexity? In this article, we present an efficient algorithm for transforming certain families of measurement-based computations into a reasonable unitary circuit model, in particular without employing the principle of deferred measurement.

1 Introduction

The one-way measurement model of quantum computation (1WQC), presented by [1] and subsequently elaborated in [2, 3], allows descriptions of unitary transformations in terms of operations, such as measurements, which are not themselves unitary. This may be regarded as being due to an appropriate combination of the observables which are measured on the one hand, and the stabilizer group of the state-space (in the “quantum-to-quantum” setting where we admit an input subsystem which may support an arbitrary input state) on the other hand. This intuition may be used in certain cases to prove that particular measurement-based computations perform unitary transformations [4], using a slight modification of the stabilizer formalism [5].

As we are often interested in describing computation in terms of unitary evolutions, measurement-based computing presents us with natural decision problems. Is it possible to efficiently determine whether a 1WQC procedure performs a unitary transformation? Is it possible to describe precisely *how* such computations transform quantum

*niel.debeaudrap@gmail.com

states, by translation to *e.g.* a unitary circuit of bounded complexity? If not, is it at least possible to efficiently verify whether it performs a *given* transformation?

These questions arise because simple unitary circuit models — such as uniform circuit families generated over the gate-set $\{H, T, \wedge Z\}$ consisting of the Hadamard gate, the $\pi/8$ phase-gate, and controlled- Z gate respectively (with no classically controlled operations and measurement deferred to the end of a computation) — are the *de facto* standard models of quantum computation. Such circuits provide the (thus far) best-developed set of idioms for uniformly expressing algorithms *independently of architecture*. Other mathematical representations of quantum computation may be useful as abstractions of proposals for implementations of quantum computers, as in the case of the measurement-based models [6, 1] and the adiabatic model [7]; and “purely physical” idioms seemingly divorced from uniform circuit families may suggest new quantum algorithms, as in the case of the $O(\sqrt{N})$ -time algorithm for evaluating NAND-trees by quantum walks [8]. However, simple unitary circuit models are currently the preferred means for expressing quantum computations.

Because of the present uncertainty as to *which* implementation proposal will prove successful, the unitary circuit model represents a natural candidate for an intermediate language for representing algorithms independently of the target architecture, even if alternative models suggest new algorithms. Therefore, one may argue that how effectively one can translate from a given model of computation to unitary circuit models, and back, is an important measure of how well-understood the model is. Unless some one proposal for implementation comes to dominate over the others, or an unlikely breakthrough (*e.g.* the discovery of an efficient classical algorithm for simulating quantum circuits) is made, translation to a “simple” model of quantum computation such as the unitary circuit model is the best that can be reasonably hoped for as a *uniform* way of obtaining low-level semantics of a quantum computation.

We are generally not interested in *arbitrary* translations to and from unitary circuit models: we should require a translation scheme which does not substantially increase the complexity of the operation being translated. For instance, provided a 1WQC procedure \mathfrak{P} for performing a unitary transformation, we may decompose each classically controlled measurement into a classically controlled change-of-basis rotation followed by a standard basis measurement; applying the principle of delayed measurement, we may then replace the classically controlled rotations with coherently controlled ones. We may then decompose the gates of this circuit with respect to a desired gate-set, such as $\{H, T, \wedge Z\}$. However, in constructions for 1WQC procedures such as those defined in [2, 3], each single-qubit transformation requires at least one additional qubit, one two-qubit entangling operation, and a classically controlled measurement which may depend on the results from several measurements. If one applies the principle of deferred measurement to such a 1WQC procedure \mathfrak{P} , which “represents” a unitary circuit \mathcal{C} using one of these constructions, the result is a circuit \mathcal{C}' which requires many more qubits than \mathcal{C} , and has at least one additional two-qubit operation for each single-qubit operation in \mathcal{C} . This represents a constant factor inflation in both the number of operations of the circuit; and depending on the ratio of single-qubit to either the number of qubits or the number of two-qubit operations in the original circuit \mathcal{C} , the asymptotic increase in the either the memory required or the number of two-qubit gates in \mathcal{C}' over \mathcal{C} is in principle unbounded.

In contrast, we may consider translation procedures \mathcal{S} from the 1WQC model, possibly with a restricted domain, which yield circuits of “reasonable” complexity. In this case, we may establish a reference for the complexity of the resulting circuits by considering (as in the previous paragraph) 1WQC procedures which arise from a translation routine \mathcal{R} from unitary circuits to the 1WQC model. We may then require that $\text{img}(\mathcal{R}) \subseteq \text{dom}(\mathcal{S})$, and that $\mathcal{S} \circ \mathcal{R}$ maps each circuit in $\text{dom}(\mathcal{R})$ to one which is no more complex (and which performs the same unitary transformation). In particular, we will be interested in the case where $\text{dom}(\mathcal{S})$ contains a wide range of operations. In this article, we require that $\text{dom}(\mathcal{R})$ consists of the set of unitary bijections generated over $\{H, T, \wedge Z\}$ (without measurements or classical control). Then $\text{dom}(\mathcal{S})$ must be similarly general, as it must “recognize” different representations of arbitrary unitary bijections in the 1WQC model, and produce unitary circuits which are equivalent both in function and efficiency to the “original”. We encapsulate these requirements as follows:^a

Definition 1.1. Let $\mathcal{C}_{\mathcal{B}}$ be a model of unitary circuits generated over a gate-set \mathcal{B} . A *semantic map for the 1WQC model in terms of unitary circuits* is an efficiently computable partial function \mathcal{S} from 1WQC procedures to $\mathcal{C}_{\mathcal{B}}$, for which membership in $\text{dom}(\mathcal{S})$ is efficiently decideable, and for which there exists a corresponding efficient translation routine \mathcal{R} from $\mathcal{C}_{\mathcal{B}}$ back to the 1WQC model such that

- $\text{dom}(\mathcal{R}) = \mathcal{C}_{\mathcal{B}}$ and $\text{img}(\mathcal{R}) \subseteq \text{dom}(\mathcal{S})$; and
- for any circuit $C \in \mathcal{C}_{\mathcal{B}}$, $C' = (\mathcal{S} \circ \mathcal{R})(C)$ performs the same transformation as C , and has complexity bounded above by that of C : in particular,
 - C' acts on at most as many qubits as C ;
 - C' has at most the same depth as C ; and
 - the number of one-qubit and two-qubit gates in C' are each at most the number of such gates in C .

We say that \mathcal{R} is a *representation map* of $\mathcal{C}_{\mathcal{B}}$ in the 1WQC model, for which \mathcal{S} provides the semantics.

Summary of results. We present a single semantic map \mathcal{S} with respect to two related constructions for 1WQC procedures: one based on the work of Raussendorf, Browne, and Briegel [2] (which we will refer to as the *simplified RBB construction*) for producing cluster-state based procedures; and the simplified procedure defined by Danos, Kashefi, and Panangaden [3] (which we will refer to as the *DKP construction*) for producing 1WQC procedures whose entanglement graphs are not required to be subgraphs of rectangular grids. We will restrict our attention to the case of unitary bijections, *i.e.* unitary circuits generated over $\{H, T, \wedge Z\}$ which do not introduce fresh qubits. In doing so, we present the following:

^aThe following definition is a refinement of the conditions described in pages 81–82 of [9] to include pairs of maps \mathcal{S} and \mathcal{R} for which, for some 1WQC procedures \mathfrak{P} , the procedure $(\mathcal{R} \circ \mathcal{S})(\mathfrak{P})$ differs in complexity from \mathfrak{P} .

A simple alternative notation for quantum circuits.

In order to facilitate descriptions of the construction of 1WQC procedures, as well as the task of determining when two circuits are congruent, we define in Section 2.1.3 a variant notation for unitary circuits (and products of linear operators generally) having convenient combinatorial properties.

An explicit account of two constructions of 1WQC procedures.

In order to explicitly describe a semantic map, we present in Section 2.3 the construction of [10] (“the DKP construction”) and a simplified version of the construction of [2] (“the simplified RBB construction”) for 1WQC procedures, including an explicit account of the time complexity of these constructions.

A succinct characterization of measurement dependencies in these constructions.

Using adjacency matrices of directed graphs, we characterize the classical measurement dependencies of the 1WQC procedures arising from the DKP construction and the simplified RBB construction in Section 3.1.

Circuit constructions from combinatorial decompositions.

In Section 3.3, we illustrate how we may efficiently decompose the structures underlying a 1WQC procedure obtained from the DKP or simplified RBB constructions. Using these decompositions, we show in Section 3.4 how such a decomposition may be used to describe unitary circuits which may perform the same operation as the original 1WQC procedure.

These results allow us to define an efficient algorithm for computing a semantic map \mathcal{S} which produces circuits for unitary bijections from an efficiently identifiable class of 1WQC procedures.

Related work.

A study into interpreting the effect of 1WQC procedures in terms of unitary circuits is performed by Childs, Leung, and Nielsen [11] via single-qubit “teleportation-based quantum computation”. Their results may be described as providing alternative proofs of validity of the constructions of Raussendorf, Browne, and Briegel [2] of particular unitary operations in the 1WQC model, but do not present conditions for when such a reduction to the circuit model may be performed.

The first algorithm to convert 1WQC procedures to unitary circuits was presented by Kashefi and Danos [12], who investigated conditions under which 1WQC procedures produced by the DKP construction perform unitary transformations. These conditions are formulated in terms of combinatorial structures, which were shown to be efficiently detectable in [13, 14]. Using these structures, [12] outlines a procedure to generate a unitary circuit using those structures. Building on the results of [13, 15, 14], we extend those results to include procedures produced by the simplified RBB construction. We also characterize the measurement dependencies which arise in such 1WQC procedures when they are in a particular canonical form, in order to verify whether such a 1WQC procedure performs the unitary described by the constructed

circuit; and demonstrate that the complexity of the circuits which are constructed are no more than the complexity of the original circuits.

The results here represent an extension of results in Chapter 3 of [9] concerning the DKP construction, by proving remarks made briefly there concerning the simplified RBB construction. In the few places where the terminology here differs from [9] (or from prior results), the differences are explicitly noted.

2 Preliminaries

We begin by presenting conventions and notational devices used throughout the article. Among these are a concise operational notation for procedures in the 1WQC model, and a transparent notation for unitary circuits which has convenient combinatorial properties. These will prove instrumental to simplify the description of the constructions for 1WQC procedures in the following section.

We take for granted a number of basic concepts in graph theory; the interested reader may refer to Diestel's text [16] for elementary definitions and proofs.

2.1 Unitary circuits

We will consider a particular model of quantum circuits: circuits without classical control, and with measurement (and trace-out operations) deferred to the end. We refer to such circuits as *unitary* circuits. As is common practise, we will ignore the terminal measurements and trace-out operations, and consider only the component involving unitary transformations and unitary embeddings.

2.1.1 Sets of approximately universal gates

As we noted in the introduction, we must consider a particular unitary circuit model with a fixed set of unitary gates. We consider a model closely related to most common gate sets: the set $\{H, T, \wedge Z\}$ consisting of the Hadamard gate H , the $\pi/8$ gate T , and the controlled- Z gate $\wedge Z$. These are expressed in matrix form by

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad T = \begin{bmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{bmatrix}, \quad \wedge Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad (2.1)$$

respectively. As H and T generate (up to scalar factors) a dense subgroup of the unitary group $U(2)$, and as $\text{CNOT} = (\mathbb{1} \otimes H) \wedge Z (\mathbb{1} \otimes H)$, the set of gates $\{H, T, \wedge Z\}$ is approximately universal for quantum computation.

To describe computations in the 1WQC model, we will also be interested in two related sets of unitary gates, involving operators such as $J(\theta)$ and $\tilde{Z}z$ given by

$$J(\theta) = H e^{-iZ\theta/2} = \frac{1}{\sqrt{2}} \begin{bmatrix} e^{-i\theta/2} & e^{i\theta/2} \\ e^{-i\theta/2} & -e^{i\theta/2} \end{bmatrix}, \quad \text{and} \quad (2.2a)$$

$$\mathcal{Z}z = e^{-i\pi Z \otimes Z/4} = \begin{bmatrix} e^{-i\pi/4} & 0 & 0 & 0 \\ 0 & e^{i\pi/4} & 0 & 0 \\ 0 & 0 & e^{i\pi/4} & 0 \\ 0 & 0 & 0 & e^{-i\pi/4} \end{bmatrix}. \quad (2.2b)$$

Note in particular that $J(\frac{m\pi}{4}) = HT^m$, and $\mathcal{Z}z = e^{i\pi/4} (T^2 \otimes T^2) \wedge Z$. Therefore, the gate-sets

$$\mathcal{B}_{\text{DKP}} = \{J(\theta), \wedge Z\}_{\theta \in \frac{\pi}{4}\mathbb{Z}} \quad \text{and} \quad \mathcal{B}_{\text{RBB}} = \{J(\theta), \mathcal{Z}z\}_{\theta \in \frac{\pi}{4}\mathbb{Z}} \quad (2.3)$$

are also approximately universal for quantum computation.^b

In unitary circuit models, one usually permits the introduction of fresh qubits into the circuit. In order to prevent the preparation of computationally expensive initial states, we will require that all fresh qubits be initially in some fixed pure state independent of the other qubits. Without loss of generality, to obtain a simpler correspondence to the operations of the 1WQC model, we will suppose that all fresh qubits are initially prepared in the $|+\rangle \propto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ state.

2.1.2 Parsimonious sets of gates

As well as being approximately universal for quantum computation, the gate sets described above have another useful property:

Definition 2.1. A set S of unitary operations is *parsimonious* if every multi-qubit gate in S is diagonal, and if distinct gates in S acting on a common qubit commute if and only if they are diagonal.^c

Each of the gate-sets described above are parsimonious. For instance, it is immediately clear that $\{H, T, \wedge Z\}$ is parsimonious; that \mathcal{B}_{DKP} and \mathcal{B}_{RBB} are parsimonious follows from the fact that

$$\begin{aligned} J(\alpha)^\dagger J(\beta)^\dagger J(\alpha) J(\beta) &= e^{iZ\alpha/2} H e^{iZ\beta/2} H^2 e^{-iZ\alpha/2} H e^{-iZ\beta/2} \\ &= e^{iZ\alpha/2} e^{iX(\beta-\alpha)/2} e^{-iZ\beta/2}, \end{aligned} \quad (2.4)$$

which is equal to $\mathbb{1}_2$ if and only if $\alpha = \beta$.

The motivation for this terminology is that, for a “parsimonious” gate-set S and for any orthogonal basis $\{|\phi_0\rangle, |\phi_1\rangle\} \in \mathcal{H}_2$ other than the standard basis, there is at most

^bThe gates $J(\pi/4)$ and $J(-\pi/4)$ alone generate a dense subgroup of $U(2)$ up to scalar factors: this is in fact shown by [17] (Section 4.5.2) in the proof that H and T are approximately universal for single-qubit operations. We include gates $J(\theta)$ for each θ an integer multiple of $\pi/4$ for both \mathcal{B}_{DKP} and \mathcal{B}_{RBB} in order to obtain a closer correspondence with circuits generated over $\{H, T, \wedge Z\}$ in the construction of 1WQC procedures in later sections.

^cThe title of [3] predates this definition of “parsimonious” by four years, and does not refer to the concept we have defined here. However, the sets of unitary operations $\{J(\theta), \wedge Z\}_{\theta \in \mathbb{R}}$ and $\{J(\theta), \wedge Z\}_{\theta \in \frac{\pi}{4}\mathbb{Z}}$ described there are both parsimonious in this sense.

one gate in S for which the vectors $|\phi_j\rangle$ are eigenvectors. Requiring that a gate-set be parsimonious is a strong constraint: by definition, operations in a unitary circuit over such a set commute if and only if they are of the same “type” (they are either equal or inverse to each other), if they act on disjoint sets of qubits, or if they are both diagonal. This is evidently useful in determining when two quantum circuits are equivalent up to transposition of gates.

2.1.3 Stable-index tensor notation for unitary circuits

For a representation map \mathcal{R} from unitary circuits to 1WQC procedures and a semantic map \mathcal{S} , to show that $\mathcal{S} \circ \mathcal{R}$ maps unitary circuits to equivalent circuits of the same (or lesser) complexity, we may attempt to show that $\mathcal{S} \circ \mathcal{R}$ maps each circuit in $\text{dom}(\mathcal{R})$ to one which is congruent^d to the original (possibly up to a set of “trivial” simplifications). For this purpose, it will be useful to have a notation for unitary circuits which is easy to produce by an algorithm, in which irrelevant details such as the order of commuting gates are essentially absent. In general, this is a problematic task: however, we will be satisfied with a notation which solves this problem for parsimonious gate-sets, such as the sets of gates which we use in this article.

One solution is to describe a notation whose meaning is invariant under permutation of any of the terms. An example of such a notation is Einstein summation notation ([18], Section B5), in which (for unitary circuits on qubits) operations U are described as tensors, using their coefficients with respect to the standard basis on individual qubits involved in the circuit. We use superscripts for column-indices, and subscripts for row-indices: thus, we have

$$H_k^j = \frac{1}{\sqrt{2}}(-1)^{jk}, \quad (2.5a)$$

$$J(\theta)_b^a = \frac{1}{\sqrt{2}}(-1)^{ab} e^{i(2b-1)\theta/2}, \quad (2.5b)$$

$$\wedge Z_{t,u}^{r,s} = \delta_{r,t} \delta_{s,u} (-1)^{rs}, \quad (2.5c)$$

and so on (where $\delta_{j,k}$ is the Kronecker delta). In an expression involving multiple gates, two such tensors are identified as acting on a common qubit when the column-index of one agrees with the row-index of another: the coefficients of the compound tensor are computed by summing over the range of the repeated index. The order of the multiplication of the tensors is then determined by the location of the index: an operation U strictly precedes another operation V if one of the row-indices of U matches a column-index of V . To represent tensor products of two operators, it suffices to juxtapose them with disjoint sets of indices. For example, given

$$C = (\mathbb{1} \otimes W) (V \otimes \mathbb{1} \otimes \mathbb{1}) U (\mathbb{1} \otimes T \otimes \mathbb{1}) (\mathbb{1} \otimes |\phi\rangle \otimes |\psi\rangle) \quad (2.6a)$$

for a generic three-qubit operator U , two-qubit W , single-qubit operators V and T , and

^dFor two unitary circuits, or two 1WQC computations, we will say that the computations are congruent if they differ only by transpositions of commuting operations (and by a possible relabelling of the qubits operated upon).

state-vectors $|\phi\rangle, |\psi\rangle \in \mathcal{H}_2$, we may equivalently write

$$C_{a_0}^{a_2, b_3, c_2} = W_{b_2, c_1}^{b_3, c_2} V_{a_1}^{a_2} U_{a_0, b_1, c_0}^{a_1, b_2, c_1} T_{b_0}^{b_1} |\phi\rangle^{b_0} |\psi\rangle^{c_0}. \quad (2.6b)$$

Furthermore, the coefficients described on the right-hand side remain the same under an arbitrary permutation of the factors, and the equality remains true after an application of any (one-to-one) relabelling of the indices.

However, while the order of the factors is not significant in Einstein notation, circuits which are equivalent up to rearrangements of commuting operations still give rise to tensor expressions which are not congruent via permutations of factors and index relabelling. This is because the tensor indices themselves fix an order of multiplication: for instance, although the operations $\wedge_{Z_{a,b}}$, $\wedge_{Z_{a,c}}$, and $\wedge_{Z_{b,c}}$ commute (where the subscripts here indicate that the operations are performed on some pair chosen from three qubits named a , b , and c), the products

$$\wedge_{Z_{b,c}} \wedge_{Z_{a,c}} \wedge_{Z_{a,b}} \equiv \wedge_{b_1, c_1}^{b_2, c_2} \wedge_{a_1, c_0}^{a_2, c_1} \wedge_{a_0, b_0}^{a_1, b_1}, \quad \text{and} \quad (2.7a)$$

$$\wedge_{Z_{a,c}} \wedge_{Z_{b,c}} \wedge_{Z_{a,b}} \equiv \wedge_{a_1, c_1}^{a_2, c_2} \wedge_{b_1, c_0}^{b_2, c_1} \wedge_{a_0, b_0}^{a_1, b_1} \quad (2.7b)$$

give rise to expressions in Einstein notation which cannot be made the same by a relabeling of indices and re-ordering of terms.

In the case that two gates commute if and only if they are both diagonal, as in circuits generated from parsimonious sets of gates, we may resolve this problem as follows. We use a notation similar to Einstein notation, but in which an index can be repeated multiple times without summation, when the index represents a qubit in a part of a unitary circuit in which the standard basis is preserved for that qubit.^e

Stable index notation. Consider a unitary operator U on N qubits, whose eigenvectors are all of the form $|s_1\rangle \otimes \cdots \otimes |s_n\rangle \otimes |\Psi\rangle$ for $s_1, \dots, s_n \in \{0, 1\}$ and $|\Psi\rangle \in \mathcal{H}_2^{\otimes m}$, where $N = n + m$. We may represent the non-zero matrix coefficients of U in the standard basis by

$$\begin{aligned} U \left[\begin{smallmatrix} s_1, \dots, s_n & y_{n+1}, \dots, y_N \\ x_{n+1}, \dots, y_N \end{smallmatrix} \right] &= \langle s_1, \dots, s_n, y_{n+1}, \dots, y_N | U | s_1, \dots, s_n, x_{n+1}, \dots, x_N \rangle \\ &= U_{s_1, \dots, s_n, x_{n+1}, \dots, x_N}^{s_1, \dots, s_n, y_{n+1}, \dots, y_N}. \end{aligned} \quad (2.8a)$$

(By hypothesis, the coefficients of U are zero when the values of the first n row- and column-indices differ.) That is, for tuples $\mathbf{s} \in \{0, 1\}^n$ and $\mathbf{a}, \mathbf{d} \in \{0, 1\}^m$, we define $U \left[\begin{smallmatrix} \mathbf{s} & \mathbf{a} \\ \mathbf{d} \end{smallmatrix} \right]$ by the operator equality

$$U = \sum_{\substack{\mathbf{s} \in \{0, 1\}^n \\ \mathbf{a}, \mathbf{d} \in \{0, 1\}^m}} U \left[\begin{smallmatrix} s_1, \dots, s_n & a_1, \dots, a_m \\ d_1, \dots, d_m \end{smallmatrix} \right] \left(|\mathbf{s}\rangle \langle \mathbf{s}| \otimes |\mathbf{a}\rangle \langle \mathbf{d}| \right). \quad (2.8b)$$

^eIf one interprets tensor indices as a random variables ranging over $\{0, 1\}$, whose value may be realized by a standard basis measurement, this corresponds to re-using indices when the value of that variable would be unchanged (or “stable”) under some number of subsequent operations.

Terminology and composition conventions for stable-index notation. In an expression $U\left[\begin{smallmatrix} \mathbf{a} \\ \mathbf{s} \mathbf{d} \end{smallmatrix}\right]$ as defined by (2.8b), we call the indices s_j *stable*, the indices d_j *deprecated*, and the indices a_j *advanced*. A product of such expressions is well-formed if each index is advanced at most once and deprecated at most once; there is no limit to the number of times an index may occur as a stable index. An index which is advanced in one term and deprecated in another is a *bound* index; otherwise it is *free*. We sum implicitly over all bound indices in a stable-index tensor expression, as in Einstein notation: we then form composite tensors $V \circ U$ by matching advanced *or* stable indices of U to corresponding deprecated *or* stable indices of V . (If the indices of U and V are disjoint, this is a tensor product of U and V .)

Example 1. The $\pi/8$ operation T is a single-qubit diagonal operator, and so has one stable index and no deprecated or advanced indices; the Hadamard operation H is a single-qubit non-diagonal operator, and so has no stable indices and one deprecated/advanced index each. We may then write their coefficients as

$$T[x] = e^{(2x-1)i\pi/8} \quad \text{and} \quad H\left[\begin{smallmatrix} v \\ u \end{smallmatrix}\right] = \frac{1}{\sqrt{2}}(-1)^{uv}. \quad (2.9a)$$

The $J(\pi/4)$ operation may be defined by composing these two operators, while the inverse of $J(\pi/4)$ is the other composition of these operators; we may write these compositions as

$$J(\pi/4)\left[\begin{smallmatrix} y \\ x \end{smallmatrix}\right] = H\left[\begin{smallmatrix} y \\ x \end{smallmatrix}\right] T[x] \quad \text{and} \quad J(-\pi/4)^\dagger\left[\begin{smallmatrix} y \\ x \end{smallmatrix}\right] = T[y] H\left[\begin{smallmatrix} y \\ x \end{smallmatrix}\right]. \quad (2.9b)$$

Finally, the tensor product $P = T \otimes H$ has eigenvectors of the form $|z\rangle \otimes |\psi\rangle$ for $z \in \{0, 1\}$ and $|\psi\rangle \in \mathcal{H}_2$ an eigenvector of H : its coefficients may then be written as $P\left[\begin{smallmatrix} z \\ y \\ x \end{smallmatrix}\right] = H\left[\begin{smallmatrix} y \\ x \end{smallmatrix}\right] T[z]$.

Identified systems. For operators $U : \mathcal{H}_2^{\otimes N} \longrightarrow \mathcal{H}_2^{\otimes N}$ acting on distinguishable qubits, we describe how U acts on these qubits by fixing an order of the qubits, where the advanced indices and deprecated indices of the qubits are presented in the same order.

Extension to general linear operators. For operators $A : \mathcal{H}_2^{\otimes N} \longrightarrow \mathcal{H}_2^{\otimes M}$ where N and M may differ, a stable-index tensor expression for A will have advanced indices which do not have corresponding deprecated indices if $M > N$, or deprecated indices which do not have corresponding advanced indices if $N > M$. Then A describes an operation which either *adds* qubits to the input space, or *discards* qubits from the input space. (These may occur when A describes an isometric embedding, or a single-qubit measurement with a selected outcome, respectively.) We may represent a qubit which is added by an advanced index with no corresponding deprecated index (padding the sequence of deprecated indices by a space, dot, or similar placeholder), and similarly for deprecated indices representing qubits which are discarded.

Example 2. We may transcribe the operator expression given in (2.6b) as

$$C\left[\begin{smallmatrix} a_2, b_3, c_2 \\ a_0, \cdot, \cdot \end{smallmatrix}\right] = W\left[\begin{smallmatrix} b_3, c_2 \\ b_2, c_1 \end{smallmatrix}\right] V\left[\begin{smallmatrix} a_2 \\ a_1 \end{smallmatrix}\right] U\left[\begin{smallmatrix} a_1, b_2, c_1 \\ a_0, b_1, c_0 \end{smallmatrix}\right] T\left[\begin{smallmatrix} b_1 \\ b_0 \end{smallmatrix}\right] |\phi\rangle\left[\begin{smallmatrix} b_0 \\ \cdot \end{smallmatrix}\right] |\psi\rangle\left[\begin{smallmatrix} c_0 \\ \cdot \end{smallmatrix}\right]. \quad (2.10)$$

The order of the indices is purely conventional (in the same way that the order of the indices of C in (2.6b) is purely conventional); we could equally well define C above as $C\left[\frac{b_3, a_2, c_2}{\cdot, a_0, \cdot}\right]$; the change in the order of the qubits causes a corresponding change in the matrix representation of C . The adjoint of the tensor in (2.10) is given by $C^\dagger\left[\frac{a_0, \cdot, \cdot}{a_2, b_3, c_2}\right]$.

A formal procedure for translating tensor notations such as in (2.6a), or involving operators $U_{a,b,\dots,q}$ labelled with the qubits that they act on, is easy to define: such a procedure is presented in Section A.2.1.

Example 3. Considering the same tensor C as in (2.10), in the special case that U has eigenvectors of the form $|a\rangle|b\rangle|\psi\rangle$ for $a, b \in \{0, 1\}$ and for some states $|\psi\rangle \in \mathcal{H}_2$, we may simplify the expression of C to

$$C\left[\frac{a_2, b_3, c_2}{a_0, \cdot, \cdot}\right] = W\left[\frac{b_3, c_2}{b_1, c_1}\right] V\left[\frac{a_2}{a_0}\right] U\left[a_0, b_1 \frac{c_1}{c_0}\right] T\left[\frac{b_1}{b_0}\right] |\phi\rangle\left[\frac{b_0}{a_0}\right] |\psi\rangle\left[\frac{c_0}{c_0}\right]. \quad (2.11)$$

The operators described by the expressions in (2.10) and (2.11) are identical in this case.

The following example illustrates the intuitive purpose of stable-index notation: by allowing stable indices to be repeated multiple times while remaining “free” variables, we also discard some redundant combinatorial information about the order of the product of operators.

Example 4. For generic unitary operators $U \in \text{U}(2)^{\otimes n}$, we may define the operator

$$\wedge U = |0\rangle\langle 0| \otimes \mathbb{1}_2^{\otimes n} + |1\rangle\langle 1| \otimes U. \quad (2.12a)$$

Let $V \in \text{U}(2)$ and $W \in \text{U}(2)^{\otimes 2}$. For qubits a, b, c , and d , consider the circuits $\wedge V_{a,d} \wedge W_{a,b,c}$ and $\wedge W_{a,b,c} \wedge V_{a,d}$ (where the subscripts here denote the qubits on which the operators act). We may represent these two circuits with stable-index expressions

$$\wedge V\left[a \frac{d_1}{d_0}\right] \wedge W\left[a \frac{b_1, c_1}{b_0, c_0}\right] \quad \text{and} \quad \wedge W\left[a \frac{b_1, c_1}{b_0, c_0}\right] \wedge V\left[a \frac{d_1}{d_0}\right], \quad (2.12b)$$

respectively. Note that these are equivalent expressions, up to permutations of the factors: this may be regarded as being due to the fact that the operations $\wedge V_{a,d}$ and $\wedge W_{a,b,c}$ commute (i.e. the circuits are congruent).

In contrast to Einstein notation, while the numerical meaning of a stable-index tensor expression is invariant under permutations of the terms, information about the order of consecutive operations is not represented when these operations preserve the standard basis of qubits on which they both act. Circuits which are congruent up to the re-ordering of commuting gates may then give rise to “equivalent” stable-index tensor expressions in the following sense:

Definition 2.2. An isomorphism of stable-index tensor expressions S_1 and S_2 is a bijective relabelling λ of the index labels of one stable-index expression S_1 to those of S_2 , and a bijective mapping τ of terms of S_1 to those of S_2 , such that for any term

$U[s_1, \dots, s_n \frac{a_1, \dots, a_m}{d_1, \dots, d_m}]$ in S_1 (with $n \geq 0$ stable indices, and $m \geq 0$ each of deprecated and advanced indices including placeholders), there is a term in S_2 given by

$$\tau \left(U[s_1, \dots, s_n \frac{a_1, \dots, a_m}{d_1, \dots, d_m}] \right) = U[\lambda(s_1), \dots, \lambda(s_n) \frac{\lambda(a_1), \dots, \lambda(a_m)}{\lambda(d_1), \dots, \lambda(d_m)}] \quad (2.13)$$

(where placeholders on the left are mapped by λ to placeholders on the right).^f

Isomorphic stable-index expressions represent not only equivalent unitary operations (which follows from the implicit summation convention), but congruent circuit decompositions as well:

Theorem 2.1. *Let C_1 and C_2 be two sequences of unitary operators over a common gate set, and acting on the same number of qubits. If the stable-index tensor representations of C_1 and C_2 are isomorphic, then C_1 and C_2 are congruent via a re-ordering of commuting operators and relabelling of the qubits. Furthermore, if C_1 and C_2 are generated from a parsimonious set of gates, the converse also holds.*

Thus, for unitary circuits constructed from a parsimonious set of gates, isomorphism of stable-index tensor expressions is equivalent to congruence up to rearrangements of commuting gates. This feature is the motivation both for this tensor notation, and for the interest in parsimonious sets of gates as in Definition 2.1. We only explicitly use advantage of this feature of this tensor notation in Section 3.5; however, other combinatorial properties of stable-index notation will frequently prove convenient in describing constructions of 1WQC procedures from the circuit model. The proof of Theorem 2.1 is provided in Section A.3.

While stable-index expressions lack any information about the order of operations which preserve the standard basis of the qubits that they act on in common, it is possible to retrieve information about the order of non-commuting gates, in order (for example) to determine how to actually perform such an operation as a physical transformation. As in Einstein notation, the order of non-commuting gates can still be determined from the indexing.

Definition 2.3. For a stable-index tensor expression S , the *index successor function* f for S is the mapping defined on the indices δ of S which are deprecated but which have a corresponding advanced index α (corresponding conventionally to the same qubit), such $f(\delta) = \alpha$ for each such δ and α .

Note that for a deprecated index $v \in \text{dom}(f)$, any term U which involves the index $f(v)$ must be performed after any operation in the circuit acting on v (excepting U itself, if it is the unique term in which v and $f(v)$ both occur). We may define:

Definition 2.4. For a stable-index tensor expression S , the *interaction hyper-graph* for S is the hypergraph G on the set of indices in S , whose edges consist of those sets of indices which are acted on by some operation in C .

^fNote that isomorphisms of stable-index representations may change the order of the terms as well as the index labels. That is, they are combinatorial homomorphisms, not just a relabelling of the indices in the terms of S_1 kept in the same sequence.

For any $v \in \text{dom}(f)$, the index $f(v)$ is adjacent to v in this hypergraph. We may then use f and G to establish a partial order $u \leq v$ on the indices of S , which describes when an index u is deprecated before another index v , given an ordering of the terms which is consistent with an order in which the operations may be performed. We may define \leq as follows. For two indices $f(v), f(w)$ which occur in a common term, each must be deprecated strictly after the indices v and w in order for the term to describe a well-defined (acyclic) product of operators. If $v \sim w$ denotes the adjacency relation in G , we then have

$$v \leq f(v) \quad \text{and} \quad w \sim f(v) \implies v \leq w \quad (2.14)$$

for any $v \in \text{dom}(f)$. We can then describe a pre-order^g $U \preceq V$ on the operations in S , where $U \preceq V$ if U has an index u and V has an index v such that $u \leq v$. We then have $U \preceq V \preceq U$ if and only if either U and V are the same term, or different terms which act on common qubits but commute; if $U \preceq V \not\preceq U$, then U must occur strictly earlier than V . We may then obtain an order for the operations in S by finding a linear order which extends \preceq .

In the case of the gate-sets defined in Section 2.1.1, the operations all act on single qubits or pairs of qubits. The interaction hyper-graph G of (a stable-index expression for) a circuit over such a gate-set will then be a conventional graph. If we ignore single-qubit diagonal operations which add no information to the ordering of the indices, and note that the square of the two-qubit operations are in each case a product of local unitaries, we may even describe the circuit by a *simple* graph (without repeated edges or loops). This graph structure, and the ordering structures we have described above for circuits will prove convenient for analyzing constructions of 1WQC procedures.

2.2 The measurement calculus

The way in which computation is performed in the 1WQC model is very different from unitary circuits as we describe them in Section 2.1. In the latter case, computation is performed with interleaving non-commuting single and two-qubit unitaries without any classical control or intermediate measurement. In contrast, a 1WQC computation consists of an initial stage in which all of the (mutually commuting) two-qubit operations are performed, followed by a sequence of classically controlled measurements and (for procedures which produce a residual quantum state as output) classically controlled single-qubit unitaries. In this section, we briefly review the elementary operations in the 1WQC model and describe standard forms for procedures in the 1WQC model as a preliminary to describing 1WQC constructions for unitary transformations, essentially following the terminology and notation introduced by Danos, Kashefi, and Panangaden [10].

^gA *pre-order* \preceq is a binary relation which is reflexive ($x \preceq x$ for all x) and transitive (if $x \preceq y$ and $y \preceq z$ then $x \preceq z$), but with no further requirements. Partial orders and equivalence relations are both examples of pre-orders. An example of a pre-order which is neither an equivalence relation nor a partial order is the relation $x \preceq y \iff \text{Re}(x) \leq \text{Re}(y)$ for complex numbers x and y .

2.2.1 Conventions for writing CPTP maps

In the 1WQC model, classical information is obtained via measurements which may in practise be destructive (*e.g.* as with photodetectors). Thus, it is usually preferable to describe 1WQC procedures by completely positive trace-preserving (CPTP) maps acting on density operators, where the input and output spaces of these operators may not be supported by the same set of qubits.

In order to better distinguish CPTP maps from linear operators on $\mathcal{H}_2^{\otimes N}$, we will represent CPTP maps in either sans-serif (as in J) or fraktur (as in \mathfrak{J}) typeface. We also adopt the following additional notational devices:

Measurement results.

When a qubit v has been measured, we will represent the result by a bit $s[v] \in \{0, 1\}$ whose value is only defined after the measurement.

Operands of CPTP maps.

The qubits on which an operation $\text{Op}_{v,w,\dots}$ acts, whether to introduce it to the system by preparation, discard it by measurements, or to transform it unitarily, are represented in the subscript. (The bit-registers $s[v]$ storing measurement results are typically omitted from such lists of operands.)

Parameterization of CPTP maps.

Operations $\text{Op}^{\alpha,\beta,s[v],s[w],\dots}$ in a 1WQC procedure will often be determined by fixed parameters α, β, \dots and the results $s[v], s[w], \dots$ of measurements on qubits v, w, \dots . (Typically, fixed parameters α denote the values of angles in the range $(-\pi, \pi]$, and operations will depend on the parities $s[v] + s[w] + \dots$ of measurement results.) Such parameters are always described in the superscript.

Finally: as there is little risk of confusion, we will typically abbreviate a composite CPTP map $\text{Op}^{(1)} \circ \text{Op}^{(2)} \circ \dots \circ \text{Op}^{(n)}$ by $\text{Op}^{(1)} \text{Op}^{(2)} \dots \text{Op}^{(n)}$.

2.2.2 Elementary operations of the 1WQC model

As in unitary circuit models, we define 1WQC computations as compositions of elementary operations. We define these following [10].

The initial stage of a 1WQC computation is the preparation of a large entangled state, which may be obtained by performing entangling operations on a collection of qubits prepared in the $|+\rangle$ state. Furthermore, the entangling operation which is performed may be described as a product of $\wedge Z$ operations acting on distinct pairs of qubits. It will then be useful to define the operations

$$\text{N}_v(\rho_S) = \rho_S \otimes |+\rangle\langle+|_v \quad \text{and} \quad \text{E}_{vw}(\rho_S) = \wedge Z_{vw} \rho_S \wedge Z_{vw} \quad (2.15)$$

for a state ρ_S on a set of qubits S , where the subscripts on the right-hand sides indicate the qubits on which the operators act. For the map N_v , we require that $v \notin S$, and obtain the system $S \cup \{v\}$ as output; for E_{vw} , we require that $v, w \in S$, and obtain the same system S as output. We refer to N_v as a *preparation* map, and E_{vw} as an *entangling* map.

The single-qubit measurements we require will be with respect to orthonormal bases on “the XY plane” (of the Bloch sphere): that is, with respect to uniform superpositions of $|0\rangle$ and $|1\rangle$ which we denote by

$$|+\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle) \quad \text{and} \quad |-\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle - e^{i\theta}|1\rangle) \quad (2.16)$$

for angles $-\pi < \theta \leq \pi$. Using these states, we may define the measurement operators

$$M_v^\theta(\rho_S) = \langle +\theta | \rho_S | +\theta \rangle_v \otimes |0\rangle\langle 0|_{s[v]} + \langle -\theta | \rho_S | -\theta \rangle_v \otimes |1\rangle\langle 1|_{s[v]}, \quad (2.17)$$

acting on a density operator ρ_S on a set of qubits S . (Recall that $s[v]$ is a classical bit storing the boolean measurement result.^h) We require that $v \in S$; the output consists of the system $S \setminus \{v\}$, ignoring the newly defined bit $s[v]$.

We extend this notation in a simple way to describe how measurements are adapted according to previous measurements: we introduce the abbreviation

$$M_v^{\theta;\beta} = M_v^{(-1)^\beta \cdot \theta} \quad (2.18)$$

for a classically controlled measurement performing either M^α or $M^{-\alpha}$, depending on a boolean expression $\beta = s[a] + s[b] + \dots$ representing the parity of some set of measurement results.ⁱ We refer to θ as the *default (measurement) angle* of the operation, and β a *sign-dependency* of v ; we say that v has a *sign-dependency on w* when such an expression β depends non-trivially on $s[w]$.

For 1WQC procedures which produce a residual quantum state as output (as we consider in this article), it is also necessary to adapt the final state depending on the outcomes of the measurements by using classically controlled single-qubit operations. For boolean expressions β describing the parities of some measurement outcomes, we may decompose these corrections into X and Z operations with the operators

$$X_v^\beta(\rho_S) = X_v^\beta \rho_S X_v^\beta \quad \text{and} \quad Z_v^\beta(\rho_S) = Z_v^\beta \rho_S Z_v^\beta, \quad (2.19)$$

acting on a density operator ρ_S on a set of qubits S which contains v .

Finally, a purely classical operation which is useful in descriptions of 1WQC procedures is a classical bit-flip operation on a measurement result $s[v]$:

$$S_{s[v]}^\beta(\rho) = |\beta\rangle\langle 0|_{s[v]} \rho |0\rangle\langle \beta|_{s[v]} + |1-\beta\rangle\langle 1|_{s[v]} \rho |1\rangle\langle 1-\beta|_{s[v]}, \quad (2.20)$$

which acts on the bit $s[v]$ which stores the result of a previous measurement.^j This operation is also known as a *signal shift* operation, as it can be used to remove the

^hWhile we typically do not represent the classical bits used in operations in terms of density operators, we may do so when this is convenient for exposition.

ⁱThe calculus defined in [10] also admits the possibility of a measurement angle being modified by a possible addition of π , depending on a boolean expression (as with sign dependencies above). As this elaboration is not necessary for measurement-based simulation of unitary circuits, we omit it in order to present a streamlined description of how to construct a 1WQC procedure in “standard” form.

^jSuch “shift” operators are usually not noted in accounts of measurement-based computation (with [10] being an exception), probably because they are simple operations on classical bits. However, in any implementation of quantum computers based on single-qubit measurements, shift operators may in practise be used to describe how parity expressions β are computed in the midst of computations. Thus, while much of the analysis of this article follows the convention of eliminating shift operations where possible, it seems productive to extend the sense of “standard forms” found in the literature to include shift operations between measurements.

dependency on the result (or *signal*) of the measurement of some qubit a from the measurement of a later qubit v ; the influence of the measurement on a is then accumulated into the classical bit $s[v]$. As a result, shift operators $S_{s[v]}^\beta$ only occur after the measurement of the corresponding qubit v .

2.2.3 Elementary transformations of 1WQC procedures

A typical 1WQC procedure is a composition of many operations as described above. In composing them, we may simplify them through simple transformations of the operations involved. At the same time, we wish to bring these expressions into a canonical form in which the entangling and measuring operations are separated into different phases of the computation (one of the motivations of the 1WQC model being to separate the tasks of creating entanglement, and maintaining control of the states of individual qubits). We now describe these transformations.

A well-formed 1WQC procedure prepares each qubit at most once, in which case that preparation is the first operation performed. We may then accumulate all preparation operations N_v to the beginning of a measurement-based procedure.^k Thus, for an operation Op_S acting on a set of qubits S , we have

$$N_v \text{Op}_S = \text{Op}_S N_v, \quad \text{when } v \notin S. \quad (2.21)$$

(When $v \in S$, the expression on the left-hand side is not well-defined.) We may collect these preparation maps together and describe them as a single operation on many qubits: to this effect, we define the shorthand

$$N_{v_N} \cdots N_{v_2} N_{v_1} = N_{\{v_1, \dots, v_N\}}. \quad (2.22)$$

Similarly, because every operation aside from the preparation maps and entangling maps are either measurements (which are the last operation performed on any qubit), Pauli corrections (which we may easily transform by commutation with $\wedge Z$, a Clifford operation), or shift operators (which do not act on qubits), we may apply the following commutation operations to accumulate all of the entangling maps to the beginning of a 1WQC procedure (just after the preparation maps):

$$E_{vw} M_p^{\theta; \beta} = M_p^{\theta; \beta} E_{vw}, \quad \text{when } p \notin \{v, w\}; \quad (2.23a)$$

$$E_{vw} S_{s[p]}^\beta = S_{s[p]}^\beta E_{vw}, \quad \text{when } p \notin \{v, w\}; \quad (2.23b)$$

$$E_{vw} X_v^\beta = Z_w^\beta X_v^\beta E_{vw}, \quad \text{and similarly for } E_{vw} X_w^\beta; \quad (2.23c)$$

$$E_{vw} Z_v^\beta = Z_v^\beta E_{vw}, \quad \text{and similarly for } E_{vw} Z_w^\beta. \quad (2.23d)$$

(When $p \in \{v, w\}$, the expression on the left-hand side of (2.23a) is not well-defined, and the bit $s[p]$ on the left-hand side of (2.23b) does not have a defined value.) As $\wedge Z$ is a diagonal operation, we also have

$$E_{uv} E_{vw} = E_{vw} E_{uv}. \quad (2.23e)$$

^kComposition is performed from right to left, as with standard notations for unitary circuits; thus, the beginning of a 1WQC procedure is on the right-hand side of a sequence of operations.

As with preparation maps, we may collect these entangling maps together and describe them as a single operation on many qubits. Controlled- Z operators act symmetrically on pairs of qubits: in stable-index notation (Section 2.1.3), we have $\wedge Z[v, w] = \wedge Z[w, v]$. Then the order of the entangling operations and the order of the qubits acted on for any single entangling map is unimportant. As $\wedge Z$ is self-inverse, we have $E_{uv} E_{uv} = \mathbb{1}$; then we may represent a product of entangling maps simply by an *entanglement graph*

$$E_G = \prod_{uv \in E(G)} E_{uv} , \quad (2.24)$$

where the edge-set $E(G)$ consists of all of the pairs of qubits on which entangling maps act.

We may also consider how to commute correction operations past other operations. First, note that commuting X^β and Z^γ operations on a state vector accrues at most a sign factor. Then, the corresponding CPTP maps commute:

$$X_u^\beta Z_v^\gamma = Z_v^\gamma X_u^\beta . \quad (2.25)$$

Similarly, X operations on any two qubits commute, as do Z operations. We may accumulate corrections of the same type on a common qubit:

$$X_u^\beta X_u^{\beta'} = X_u^{\beta+\beta'} , \quad Z_u^\gamma Z_u^{\gamma'} = Z_u^{\gamma+\gamma'} . \quad (2.26)$$

With respect to measurement operations, corrections on a qubit v may be interpreted as (classically controlled) changes of basis for the measurement on v ; then, we can consider “absorbing” correction operations on qubits to be measured into the measurement operations. Consider a measurement in the $|+\theta\rangle, |-\theta\rangle$ basis for some $-\pi < \theta \leq \pi$. We have

$$X|+\theta\rangle = \frac{1}{\sqrt{2}}(e^{i\theta}|0\rangle + |1\rangle) \propto \frac{1}{\sqrt{2}}(|0\rangle + e^{-i\theta}|1\rangle) = |+(-\theta)\rangle, \quad (2.27)$$

and similarly $X|-\theta\rangle \propto |-(\theta)\rangle$. We may then incorporate X_v^β corrections into the basis of a subsequent measurement on v via the equation

$$M_v^\theta X_v^\beta = M_v^{(-1)^\beta \cdot \theta} = M_v^{\theta;\beta} \quad (2.28)$$

for a boolean expression β , using the notation defined in (2.18): more generally, we have $M_v^{\theta;\beta} X_v^\gamma = M_b^{\theta;\beta+\gamma}$. We may describe the effect of Z corrections on measurements in a similar fashion:

$$Z|+\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle - e^{i\theta}|1\rangle) \propto |-\theta\rangle, \quad (2.29)$$

and similarly $Z|-\theta\rangle = |+\theta\rangle$. For an arbitrary state ρ_S on a set of qubits S including v , we then have

$$\langle +\theta | [Z_v \rho_S Z_v^\dagger] | +\theta \rangle_v = \langle -\theta | \rho_S | -\theta \rangle_v , \quad \text{and} \quad (2.30a)$$

$$\langle -\theta | [Z_v \rho_S Z_v^\dagger] | -\theta \rangle_v = \langle +\theta | \rho_S | +\theta \rangle_v . \quad (2.30b)$$

Thus, Z corrections effectively exchange the roles of the states $|\pm_\theta\rangle$ for measurements made in that basis. As this exchange is independent of any classically-controlled change of sign in θ , we then obtain

$$M_v^{\theta;\beta} Z_v^\gamma = S_{s[v]}^\gamma M_v^{\theta;\beta} \quad (2.31)$$

for arbitrary boolean expressions β and γ . (That is, we may toggle the measurement result $s[v]$ instead of performing a Z correction on v prior to measurement.) For such a shift operator $S_{s[v]}^\gamma$ which occurs after a measurement M_v^* and before any operations using $s[v]$ for classical control, we call γ a *bit-dependency* of v .¹

There are some further simplifications to measurement operations which are possible when the measurement angle θ is a multiple of $\pi/2$. We refer to these as *Pauli simplifications*, as such measurements are with respect to the eigenbases of the X or Y Pauli operators. For $\theta \in \{0, \pi\}$, we have $|+_{(-\theta)}\rangle = |+_\theta\rangle$ and $Z|-\theta\rangle \propto |-(\theta)\rangle$; thus, we have

$$M_v^{m\pi;\beta} = M_v^{m\pi} , \quad (2.32)$$

so that we may ignore sign-dependencies in this case. For $\theta \in \{-\pi/2, \pi/2\}$, we have $X|+\theta\rangle \propto Z|+\theta\rangle$ and similarly for $|-\theta\rangle$, so that

$$M_v^{\pm\pi/2;\beta} = M_v^{\pm\pi/2} X_v^\beta = M_v^{\pm\pi/2} Z_v^\beta = S_{s[v]}^\beta M_v^{\pm\pi/2} . \quad (2.33)$$

Thus, measurements in so-called Pauli bases may be performed without dependencies on previous measurements. In summary, we have the following relations which allow correction operations to be commuted past or absorbed into measurements:

$$M_v^{\theta;\beta} X_u^\gamma = X_u^\gamma M_v^{\theta;\beta} ; \quad (2.34a)$$

$$M_v^{\theta;\beta} X_u^\gamma = X_u^\gamma M_v^{\theta;\beta} , \quad \text{when } u \neq v; \quad (2.34b)$$

$$M_v^{\theta;\beta} Z_u^\gamma = Z_u^\gamma M_v^{\theta;\beta} , \quad \text{when } u \neq v; \quad (2.34c)$$

$$M_v^{\theta;\beta} X_v^\gamma = M_v^{\theta;(\beta+\gamma)} ; \quad (2.34d)$$

$$M_v^{\theta;\beta} Z_v^\gamma = S_{s[v]}^\gamma M_v^{\theta;\beta} ; \quad (2.34e)$$

$$M_v^{m\pi;\beta} = M_v^{m\pi} ; \quad (2.34f)$$

$$M_v^{\pm\pi/2;\beta} = S_{s[v]}^\beta M_v^{\pm\pi/2} . \quad (2.34g)$$

We will occasionally be interested in describing transformations of 1WQC procedures “without Pauli simplifications”, in which case we do not use the relations given in (2.34f) and (2.34g).

Finally, we consider the transformations which are possible by commuting shift operators. We may accumulate multiple signal shifts acting on a single measurement result:

$$S_{s[v]}^\beta S_{s[v]}^\gamma = S_{s[v]}^{\beta+\gamma} , \quad (2.35)$$

¹This corresponds essentially to what are called π -dependencies in [9], and to “dependencies caused by Z -actions” in [10].

where this sum is evaluated modulo 2. If desired, we may also eliminate shift operations entirely from a 1WQC procedure, by incorporating the bit-flip which is performed on the measurement signal $s[v]$ into every expression which depends on $s[v]$. This has the effect of increasing the complexity of the classical control expressions, but when performed for all measurement results $s[v]$, also makes explicit the dependency of any operation on prior measurement results. For an arbitrary operation $\text{Op}^{f(\dots, s[v], \dots)}$ which is classically controlled by some function f taking $s[v]$ as a parameter, we have

$$\text{Op}^{f(\dots, s[v], \dots)} S_{s[v]}^\beta = S_{s[v]}^\beta \text{Op}^{f(\dots, s[v] + \beta, \dots)} ; \quad (2.36)$$

that is, we substitute $s[v] + \beta$ wherever $s[v]$ occurs, again evaluated modulo 2. We may use this commutation relation to accumulate all shift operators at the right-hand side of a 1WQC procedure, after every operation acting on qubits: we refer to this as *signal shifting*. Conventionally, we may discard any classical measurement results when they are no longer required; thus, at the right-hand end of a 1WQC procedure, we may remove any signal shift operators which remain.

2.2.4 Standard forms for 1WQC procedures

Using the simplifications above, we may transform 1WQC procedures into “standard” orderings of the operations which are comparatively convenient for discussion, and are also of low operational depth: we now describe two different such forms.

As we have noted above, in any well-formed 1WQC procedure, it is possible to commute all preparation and entangling maps to the beginning of the procedure (on the right), and to propagate all correction operations to the end (on the left), possibly absorbing corrections into measurement operations while doing so. Some of the corrections will also induce shift operations just after the measurement into which they are absorbed; and we may do this in a uniform way without explicit knowledge (or despite uncertainties in) the values of the default measurement angles. This motivates the following:

Definition 2.5. A 1WQC procedure \mathfrak{P} is said to be in *standard form*^m if it can be decomposed as

$$\mathfrak{P} = \mathbf{C} \circ \mathbf{M} \circ \mathbf{E}_G \circ \mathbf{N}_P , \quad (2.37)$$

where \mathbf{N}_P is a product of preparation maps acting on some set of qubits P , \mathbf{E}_G is a product of entangling operations acting on pairs uv connected by an edge in the graph G , \mathbf{M} consists entirely of measurement operations and shift operations (where any shift operation $S_{s[v]}^*$ immediately follows the measurement M_v^*), and \mathbf{C} consists entirely of correction operations. The procedure of transforming a 1WQC procedure into such a form using equations (2.21), (2.23), and (2.34a)–(2.34e) is called *standardization*.

Note in particular that the process of “standardization” does not apply Pauli simplifications or signal shifting. For practical applications, it is probably preferable to perform Pauli simplifications, at least; but for the purposes of discussion it is useful to

^mThis terminology is similar to, but differs slightly from, the usage in [10] and [9].

consider the case where such simplifications of expressions for 1WQC procedures are not applied.

By applying Pauli simplifications on measurements and signal shifting, it is possible to have equivalent 1WQC procedures which are each in standard form. Rather than attempt to characterize when such expressions are equal, it is useful to consider a form in which any variation of measurement dependencies has been removed. We may define such a form as follows:

Definition 2.6. A 1WQC procedure \mathfrak{P} is said to be in *normal form* if it is in standard form, does not contain any shift operations, and any measurement operator with an angle $\theta \in \frac{\pi}{2}\mathbb{Z}$ has no measurement dependencies. We call the a procedure \mathfrak{P} in normal form the *normalization* of another \mathfrak{P}' if we may obtain \mathfrak{P} from \mathfrak{P}' by standardizing \mathfrak{P}' , applying Pauli simplifications (2.34f)–(2.34g), and then performing signal shifting (2.36), removing the shift operators from the left after doing so.

While it may not be desirable to compute such expressions for the practical application of 1WQC procedures, obtaining the normalizations of 1WQC procedures will be useful for the analysis of this article.

2.2.5 Composability of 1WQC procedures

To perform universal computation, it is necessary to *compose* 1WQC computations: that is, to apply a second 1WQC procedure \mathfrak{P}_2 to the output system produced by another 1WQC procedure \mathfrak{P}_1 . Therefore, when composing 1WQC procedures $\mathfrak{P} = \mathfrak{P}_2 \circ \mathfrak{P}_1$, we will be interested in restricting the qubits which are allocated in \mathfrak{P}_2 to be distinct from any qubits involved in \mathfrak{P}_1 , except those which \mathfrak{P}_1 produces as output.

It will be convenient at this point to introduce a combinatorial structure which partially describes a 1WQC procedure in order to define composability conditions (as well as to visually convey the structures which will be important in later sections). For a 1WQC procedure \mathfrak{P} , we may consider the information contained in \mathfrak{P} which is independent of ordering or of the particular measurement angles or outcomes:

Definition 2.7. The *geometry underlying a 1WQC procedure* \mathfrak{P} is a triple (G, I, O) , where G is the entanglement graph on the qubits of \mathfrak{P} specified by the entangling maps E_{vw} in \mathfrak{P} , $I \subseteq V(G)$ is the set of qubits which are not prepared by \mathfrak{P} , and $O \subseteq V(G)$ is the set of qubits which are not measured by \mathfrak{P} . We refer to I and O as the *input subsystem* and *output subsystem* of \mathfrak{P} , respectively. (When a given graph G may be inferred from context, we may write $I^c = V(G) \setminus I$ and $O^c = V(G) \setminus O$ for the complements of these subsystems.)

We will often illustrate geometries in the style shown in Figure 1. Such diagrams give a partial description of a 1WQC procedure, in which each qubit $v \in I^c$ is prepared, the entangling operations E_{vw} implied by the edges $vw \in E(G)$ are performed, and the qubits $w \in O^c$ are measured. However, the measurement order, measurement angles and dependencies, and any correction/shift operations are left unspecified.

Using geometries, we can express conditions for the composability of two 1WQC procedures in terms of a sense in which the geometries of these patterns may be “composed”:

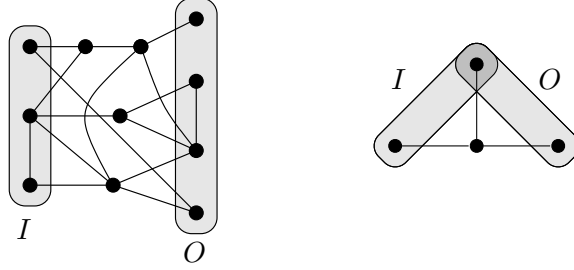


Figure 1: Illustration of two geometries for 1WQC procedures. Labels for the vertices have been omitted. Note in particular that the sets I and O may overlap (indicated by the more darkly shaded region in the geometry on the right).

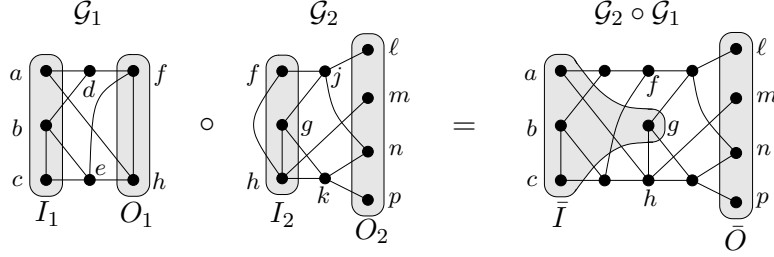


Figure 2: Illustration of the composition of two generic geometries for 1WQC procedures. Notice that composition in the diagrams is left-to-right, while the composition in written notation is right-to-left.

Definition 2.8. For two geometries $\mathcal{G}_1 = (G_1, I_1, O_1)$ and $\mathcal{G}_2 = (G_2, I_2, O_2)$, we say that \mathcal{G}_1 and \mathcal{G}_2 are *composable* if $[V(G_1) \cap V(G_2)] \subseteq [I_2 \cap O_1]$. The *composition* $\bar{\mathcal{G}} = \mathcal{G}_2 \circ \mathcal{G}_1$ is then defined by $\bar{\mathcal{G}} = (\bar{G}, \bar{I}, \bar{O})$, where

$$V(\bar{G}) = V(G_1) \cup V(G_2), \quad (2.38a)$$

$$E(\bar{G}) = E(G_1) \triangle E(G_2), \quad (2.38b)$$

$$\bar{I} = I_1 \cup (I_2 \setminus O_1), \quad (2.38c)$$

$$\bar{O} = O_2 \cup (O_1 \setminus I_2), \quad (2.38d)$$

where $A \triangle B = [A \setminus B] \cup [B \setminus A]$ is the symmetric difference of A and B . Two 1WQC procedures \mathfrak{P}_1 and \mathfrak{P}_2 are *composable* if their underlying geometries \mathcal{G}_1 and \mathcal{G}_2 are composable; the underlying geometry of the procedure $\mathfrak{P}_2 \circ \mathfrak{P}_1$ is then $\mathcal{G}_2 \circ \mathcal{G}_1$.

Figure 2 illustrates the composition of two geometries (and, by way of synecdoche, represents the composition of two 1WQC procedures).

In order to obtain a composition of two 1WQC procedures $\mathfrak{P}_2 \circ \mathfrak{P}_1$, we may relabel the qubits involved in \mathfrak{P}_2 in order for the conditions of (2.38) to hold. From this point

onwards, when we wish to compose two generic 1WQC procedures \mathfrak{P}_1 and \mathfrak{P}_2 , we will assume that they are composable in the sense above, eliding over the procedure of relabelling the qubits to be prepared which may be necessary.

2.3 The DKP and simplified RBB constructions

Having established stable-index tensor notation (Section 2.1.3) and a calculus of the operations used in a 1WQC procedure (Section 2.2), we may describe two constructions for approximately universal quantum computation in the 1WQC model.ⁿ The first construction we will refer to as the *DKP construction*, and is based on the work of Danos, Kashefi, and Panangaden [10] for the construction of 1WQC procedures in the absence of topological constraints on the interactions of qubits. The second construction is derived from the elementary constructions among those presented by Raussendorf, Browne, and Briegel [2] for producing 1WQC procedures, under the topological constraint that the entanglement graph be an induced subgraph of a suitably large rectangular grid; we call this the *simplified RBB construction*.^o

2.3.1 Construction of the gate-sets \mathcal{B}_{DKP} and \mathcal{B}_{RBB}

The first step in both of the constructions we describe is to identify 1WQC procedures for the gate-sets $\mathcal{B}_{\text{DKP}} = \{J(\theta), \wedge Z\}_{\theta \in \frac{\pi}{4}\mathbb{Z}}$ and $\mathcal{B}_{\text{RBB}} = \{J(\theta), \mathcal{Z}z\}_{\theta \in \frac{\pi}{4}\mathbb{Z}}$. We shall do this in overview, leaving the details for Section B.1.

By definition, the CPTP entangling map E_{vw} performs controlled- Z operations on pairs of qubits: for a system S including qubits v, w , we have

$$E_{vw}(\rho_S) = \wedge Z_{vw} \rho_S \wedge Z_{vw} . \quad (2.39)$$

We may implement $\mathcal{Z}z$ and $J(\theta)$ operations in the 1WQC model with somewhat more elaborate procedures:

$$\mathfrak{Z}_{v,w} = Z_v^{s[a]} Z_w^{s[a]} M_a^{\pi/2} E_{av} E_{aw} N_a , \quad (2.40)$$

$$\mathfrak{J}_{w/v}^\theta = X_w^{s[v]} M_v^{-\theta} E_{vw} N_w , \quad (2.41)$$

where the subscript w/v in the latter denotes that it discards v from the input system and introduces a new qubit w . It is possible to show that the effects of these two 1WQC procedures are given by

$$\mathfrak{Z}_{v,w}(\rho_S) = \mathcal{Z}z_{v,w} \rho_S \mathcal{Z}z_{v,w}^\dagger , \quad (2.42)$$

ⁿThe presentation of such construction presented here differs from the approach of [10]: while the latter proceeds by defining the composition of formal denotational semantics; we will adopt a somewhat more informal approach involving stable-index tensor notation.

^oThe simplification implied in this title is twofold. Firstly, we use the $\mathcal{Z}z$ operation given in (2.2b) to describe the construction of arbitrary operations, rather than the CNOT operation as in [2], page 5. While this will lead in practise to a constant-factor reduction in the complexity of some 1WQC procedures, this construction is clearly implicit in [2], and so we may attribute this construction to that work. Secondly, the “simplified” RBB construction omits the special-purpose constructions exhibited in Sections IV.A–F of [2], such as compact patterns for reversing an block of consecutive qubits, non-nearest neighbor CNOT operations, the quantum Fourier transform over \mathbb{Z}_{2^k} for $k \geq 0$, and addition circuits for \mathbb{Z}_{2^k} .

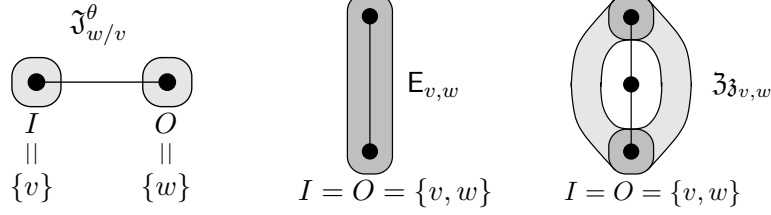


Figure 3: Illustration of the geometries of the three elementary 1WQC procedures $\mathfrak{J}_{w/v}^\theta$, E_{vw} , and $\mathfrak{Z}_{v,w}$ for generic qubits v and w , given by (2.39), (2.40), and (2.41) respectively. Note that the input and output subsystems coincide for E_{vw} and $\mathfrak{Z}_{v,w}$.

for arbitrary operators ρ_S acting on a system including v and w , and

$$\mathfrak{J}_{w/v}^\theta \left(|\Psi\rangle\langle\Psi|_S \right) = J(\theta)_{w/v} |\Psi\rangle\langle\Psi|_S J(\theta)_{v/w}^\dagger, \quad (2.43)$$

for a system S including v but not including w , where by $J(\theta)_{w/v}$ we denote the linear operator $\mathcal{H}_v \rightarrow \mathcal{H}_w$ which maps state-vectors $|\psi\rangle_v \mapsto [J(\theta)|\psi\rangle]_w$. Thus, the procedures E_{vw} , $\mathfrak{Z}_{v,w}$, and $\mathfrak{J}_{w/v}^\theta$ given in (2.39), (2.40), and (2.41) respectively may be used to represent each of the operations in \mathcal{B}_{DKP} and \mathcal{B}_{RBB} . The geometries underlying these three 1WQC procedures are illustrated in Figure 3.

2.3.2 Corresponding stable-tensor indices to qubits in the 1WQC model

By the analysis of Section 2.3.1, we have a simple correspondence between unitary circuit operations described in terms of stable-index tensor expressions, and simple 1WQC procedures for effecting them: we have

$$J(\theta)\left[\frac{w}{v}\right] \longrightarrow \mathfrak{J}_{w/v}^\theta; \quad \wedge Z[v,w] \longrightarrow E_{vw}; \quad \mathcal{Z}z[v,w] \longrightarrow \mathfrak{Z}_{v,w}. \quad (2.44a)$$

We may extend this correspondence also to account for the preparation of fresh qubits: as a preparation map N_v simply prepares a qubit v in the $|+\rangle$ state, we may also write

$$|+\rangle\left[\frac{v}{u}\right] \longrightarrow N_v. \quad (2.44b)$$

These parallels between operations on indices in stable-index expressions, and CPTP maps operations on qubits in the 1WQC model, also extend to syntactical considerations:

- If we order terms in a stable-index expression from right to left according to an order in which the operations may actually be performed, by construction there cannot be any terms acting on an index v to the right of an operation such as $J(\theta)\left[\frac{w}{v}\right]$ which deprecates v ; nor operations to the left of v before an operation such as $J(\theta)\left[\frac{v}{u}\right]$ in which v is advanced. Similarly, in a 1WQC procedure, operations acting on a qubit v are not permitted after it has been measured, nor before it has been prepared.

- In a stable-index expression for unitary circuits, stable indices (which are neither advanced nor deprecated) are those indices representing the a qubit whose configuration over the standard basis are unchanged by the operator acting on it. Similarly, in a 1WQC procedure, a qubit which is neither added to the system nor discarded is in both the input and output subsystems of the procedure, and so is only acted on by diagonal operations (setting aside possible X operations), which similarly do not affect the diagonal terms of their reduced density operators.

Thus, the role of tensor indices (in a stable-index expression) in describing quantum states as distributions over the standard basis is analogous to the role of qubits in a 1WQC procedure which actually support quantum states. We may then obtain 1WQC procedures for unitary embeddings U by simply applying the correspondences in (2.44) to a stable-index expression of a unitary circuit. While the constructions of [10] or [2] are not presented in precisely such terms, the DKP construction and simplified RBB constructions may be described as transliterating a sequence of gates in a unitary circuit into 1WQC procedures of the form \mathfrak{J}^θ , E , and \mathfrak{J}_3 , yielding a 1WQC procedure for the circuit as a whole. We present an explicit algorithm to do so in Section 2.3.

In the case of the DKP construction, in which the qubits of the resulting procedures are not subject to any topological constraints, application of the correspondence in (2.44) is an essentially complete summary of the construction. The case of the simplified RBB construction is similar, but is subject to topological constraints which we describe below.

2.3.3 Topological constraints in the simplified RBB construction

The simplified RBB construction is subject to the constraint that the entangled states which are used must be obtained from a *cluster state*: that is, a state in which qubits involved are arranged in a two-dimensional grid, and two qubits v, w are acted on with an entangling map E_{vw} if and only if v and w are nearest neighbors in the grid. We will describe circuit constructions in which the indices of the stable-index representation satisfies the same constraints as required for the qubits in the cluster-state-based variant of the 1WQC model, which will allow us to employ the correspondence of (2.44) directly.

In the cluster-state-based (and original) variant of the 1WQC model, the only tool provided for obtaining graphs *other* than a rectangular grid is the removal of qubits from such a grid, via measurement in the $|0\rangle, |1\rangle$ basis. Temporarily extending the set of operations defined in Section 2.2.2, we may define a M_v^z operation which performs a destructive single-qubit measurement of a qubit v in the standard basis, and produces a bit $s[v]$ storing the result in the obvious manner. We may then show

$$M_v^z E_{vw} = Z_w^{s[v]} M_v^z, \quad (2.45)$$

which follows from $|x\rangle_{s[v]} \langle x|_v \wedge Z_{vw} = Z_w^x |x\rangle_{s[v]} \langle x|_v$ for $x \in \{0, 1\}$. We may use this to remove all of the entangling maps E_{vw} incident to a qubit v , propagating the measurement to the right until it is performed immediately after the preparation of v . If we remove the redundant preparation and measurement of v (and the corrections

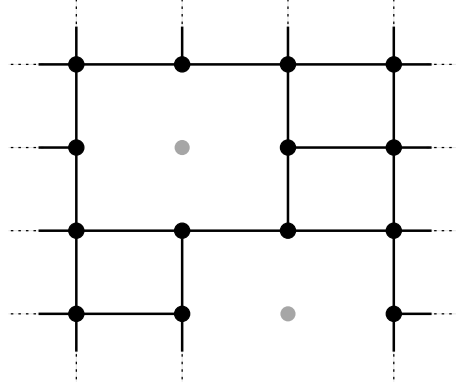


Figure 4: Illustration of the effect of removing vertices from a two-dimensional grid. This corresponds to the effect of performing measurement of select qubits in a cluster state, in the standard basis. Vertices correspond to qubits, with vertices in grey corresponding to qubits which have been measured; edges (or absence of edges) correspond to the presence (resp. absence) of entangling maps on adjacent qubits.

induced by that measurement), this procedure has the effect of removing the vertex v from the grid. This is illustrated in Figure 4.

We may instead suppose (equivalently) that we may prepare any subset of the qubits in a two-dimensional lattice (omitting those qubits which we would subsequently remove as described above), and that two qubits are subject to an entangling map if and only if they are nearest neighbors in the grid. That is: we will merely require that the entanglement graph G is an induced subgraph of a rectangular lattice. Given this restriction on 1WQC procedures, we impose the corresponding restriction that the circuits which are accepted as input must have a linear nearest-neighbor topology.

To represent products of single-qubit gates as a 1WQC procedure, in the DKP construction as well as in the simplified RBB construction, we may compose patterns of the form $\mathfrak{J}_{w/v}^\alpha$ as in

$$\mathfrak{J}_{v_N/v_{N-1}}^{\alpha_{N-1}} \cdots \mathfrak{J}_{v_2/v_1}^{\alpha_1} \mathfrak{J}_{v_1/v_0}^{\alpha_0} . \quad (2.46)$$

We may refer to such a procedure as a *chain pattern*. Note that in such a procedure, we require that the qubits v_j for $0 < j < N$ to have degree 2 in the corresponding entanglement graph, having no neighbors aside from v_{j-1} and v_{j+1} . In the simplified RBB construction, the degrees of these qubits must be “enforced” by a representation in the grid, where every neighbor of these qubits v_j in the grid (for $0 < j < N$) are removed except for the neighbors v_{j-1} and v_{j+1} . An illustration of such an embedding of a chain in the grid is illustrated in Figure 5. (Typically, such chain patterns would be embedded as a horizontal path through the grid, but as Figure 5 also shows, we may also employ more general paths in the grid.)

Two logical qubits in a circuit which do not interact must be represented by (chains of) qubits in the grid which are separated by at least two edges; otherwise there will

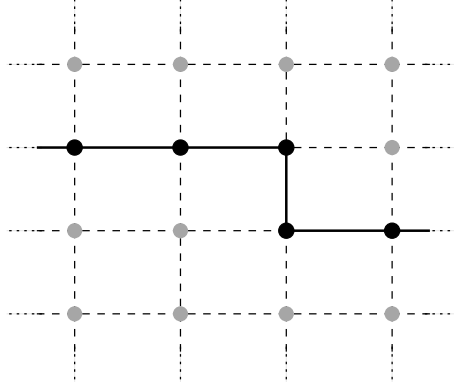


Figure 5: Illustration of an embedding of a chain in the grid. Grey dots and broken lines represent vertices and edges which are removed from the grid by vertex deletions, to produce a graph containing the illustrated sequence of vertices with degree 2.

be entangling relations between them. Thus, we will represent products of single-qubit operations on independent qubits as non-adjacent horizontal rows of the grid. We may then use a vertical column of three qubits (as illustrated in Figure 3) to implement a $\mathcal{Z}z$ operation between these non-adjacent rows, using a single qubit between the two rows to mediate the interaction. However, there are two technicalities which must be addressed:

Adjusting for uneven chain lengths. Consider a decomposition of single-qubit operators U and V on distinct qubits, as products of $J(\theta)$ operators. If the number of terms in the two products differ, the ends of the corresponding chain patterns may lie in different columns. To operate on two chains with a $\mathcal{Z}z$ operation as described above, we require that the ends of the chains representing each qubit lie in the same column. To achieve this, we may extend each chain by two or more vertices, as necessary, representing a decomposition of $\mathbb{1}$ as a product of $J(\theta)$ gates. We may define 1WQC procedures

$$\mathfrak{Jd}_{v_2/v_0}^2 = \mathfrak{J}_{v_2/v_1}^0 \mathfrak{J}_{v_1/v_0}^0, \quad \mathfrak{Jd}_{v_3/v_0}^3 = \mathfrak{J}_{v_3/v_2}^{\pi/2} \mathfrak{J}_{v_2/v_1}^{\pi/2} \mathfrak{J}_{v_1/v_0}^{\pi/2}; \quad (2.47)$$

the geometries underlying these procedures are illustrated in Figure 6. In terms of the correspondence between stable-index expressions and 1WQC procedures described in Section 2.3.2, we have

$$\mathfrak{Jd}_{v_2/v_0}^2 \longrightarrow J(0)\left[\frac{v_2}{v_1}\right] J(0)\left[\frac{v_1}{v_0}\right], \quad (2.48a)$$

$$\mathfrak{Jd}_{v_3/v_0}^3 \longrightarrow J(\pi/2)\left[\frac{v_3}{v_2}\right] J(\pi/2)\left[\frac{v_2}{v_1}\right] J(\pi/2)\left[\frac{v_1}{v_0}\right]. \quad (2.48b)$$

One can verify the stable-index products given on the right evaluate to $\mathbb{1}\left[\frac{v_2}{v_0}\right]$ and $\mathbb{1}\left[\frac{v_3}{v_0}\right]$ respectively; then both 1WQC procedures above simply transmit the state presented

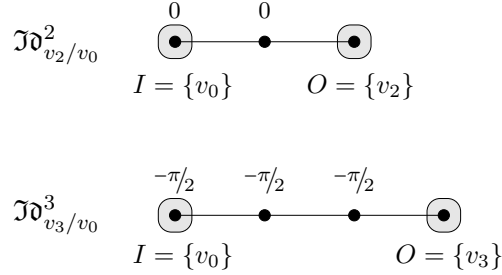


Figure 6: Illustration of the geometries underlying the procedures \mathfrak{Id}^2 and \mathfrak{Id}^3 as defined in (2.47). The default measurement angles for qubits in O^c are indicated above each qubit.

as input to its respective output system, using a chain of length either 2 or 3. By composing these procedures repeatedly, we can then implement a path of any length $\ell \geq 2$ in the grid, representing a sequence of operations which performs the identity $\mathbb{1}_2$ on a given qubit. Thus, whenever required, we may suppose that the paths in the grid corresponding to any two qubits are of the same length, regardless of the number of (non-trivial) unitary transformations performed on them.

“Forbidden” qubits adjacent to the mediating qubit in \mathcal{Zz} . Just as we require a distance of 2 between qubits in a 1WQC procedure representing non-interacting qubits in a unitary circuit, we require that the qubit a which mediates a \mathcal{Zz}_{vw} operation is adjacent only to v and w ; that is, we require that the qubits immediately to the left and right in the grid are not in the entanglement graph. Then we require a distance of two edges between any two instances of \mathfrak{Zz} performed on the same pair of rows. For instance, to represent a circuit $\mathcal{Zz}[v_f, w_f] J(\beta)[\frac{w_f}{w_i}] J(\alpha)[\frac{v_f}{v_i}] \mathcal{Zz}[v_i, w_i]$ on a pair of logical qubits v and w , in which two \mathcal{Zz} operations are separated only by a layer of single-qubit operations of depth 1, we may perform the substitution

$$\begin{aligned} & \mathcal{Zz}[v_f, w_f] J(\beta)\left[\frac{w_f}{w_i}\right] J(\alpha)\left[\frac{v_f}{v_i}\right] \mathcal{Zz}[v_i, w_i] \\ &= \mathcal{Zz}[v_f, w_f] J(0)\left[\frac{w_f}{w_2}\right] J(0)\left[\frac{w_2}{w_1}\right] J(\beta)\left[\frac{w_1}{w_i}\right] \\ & \quad J(0)\left[\frac{v_f}{v_2}\right] J(0)\left[\frac{v_2}{v_1}\right] J(\alpha)\left[\frac{v_1}{v_i}\right] \mathcal{Zz}[v_i, w_i] . \end{aligned} \quad (2.49)$$

Mapping each $J(\theta)[\frac{a}{d}]$ term to a procedure $\mathfrak{J}_{a/d}^\theta$ and each $\mathcal{Zz}[s, t]$ term to a procedure $\mathfrak{Z}_{s,t}$, the distance between the qubits v_i and v_f resulting in the right-hand side is then at least 2 (and similarly for w_i and w_f), as required.

These observations describe ways in which a 1WQC procedure for performing transformations must be “padded” by operations which perform the identity, in order to simulate a unitary circuit in the 1WQC model. We illustrate these observations

in Figure 7 as sample constructions of 1WQC procedures. However, in order to obtain a uniform description of both the DKP and simplified RBB constructions, we may instead pad the stable-index representations for circuits with the corresponding products of single-qubit unitaries as a pre-processing stage. By construction, the indices of the resulting stable-index expressions will then satisfy the same constraints which are required for the qubits in a cluster-based 1WQC computation. We may perform such padding as part of an alternative procedure for constructing stable-index expressions for circuits: such a procedure is described in Section A.2.2.

2.3.4 Unified description of the DKP and simplified RBB constructions

Using the above techniques for the simplified RBB construction, we may obtain a unified procedure describing both the DKP and simplified RBB constructions, as follows.

Procedure for both constructions. Suppose we are given a unitary circuit C over the gate-set $\{H, T, \wedge Z\}$: as described in the previous section, we further require that C have a linear nearest-neighbor topology in the case of the simplified RBB construction.

1. Obtain a circuit in normal form.

Given a unitary circuit C generated over the gate-set $\{H, T, \wedge Z\}$, we obtain an equivalent circuit as follows. We cancel pairs of $\wedge Z$ gates which act on common pairs of wires, and which are separated only by gates with which they commute (*i.e.* T operations and operations acting on other wires). In the case of the simplified RBB construction, we convert the circuit to the (essentially equivalent) gate-set $\{H, T, \mathcal{Z}z\}$ by applying the substitution

$$\wedge Z = \mathcal{Z}z(T^\dagger \otimes T^\dagger)^2 \quad (2.50)$$

to all $\wedge Z$ gates; no such translation is required for the DKP construction. We then commute all T and T^\dagger operations to the left (simplifying them using the identity $T^8 = \mathbb{1}$), collecting them in each case either at the end of the circuit, or immediately preceding a Hadamard operation the same qubit.

We call the resulting circuit C' the *normal form* of C ; this circuit consists of a product of operations $\wedge Z$ and HT^r (for various $r \in \mathbb{Z}$) for the DKP construction, and $\mathcal{Z}z$ and HT^r for the simplified RBB construction.

2. Convert the circuit to a stable-index expression in the gate-set \mathcal{B}_{DKP} or \mathcal{B}_{RBB} .

Having obtained C' , we apply the substitutions

$$HT^m = J\left(\frac{m\pi}{4}\right), \quad T^m = J(0)J\left(\frac{m\pi}{4}\right) \quad (2.51)$$

for each consecutive block of T operations, applying the right-hand equality for blocks of T operations at the end of the circuit. These substitutions yield an equivalent representation of the same operation using the gate-set \mathcal{B}_{DKP} or \mathcal{B}_{RBB} , depending on whether the circuit makes use of $\wedge Z$ or $\mathcal{Z}z$ gates. As we do so, we

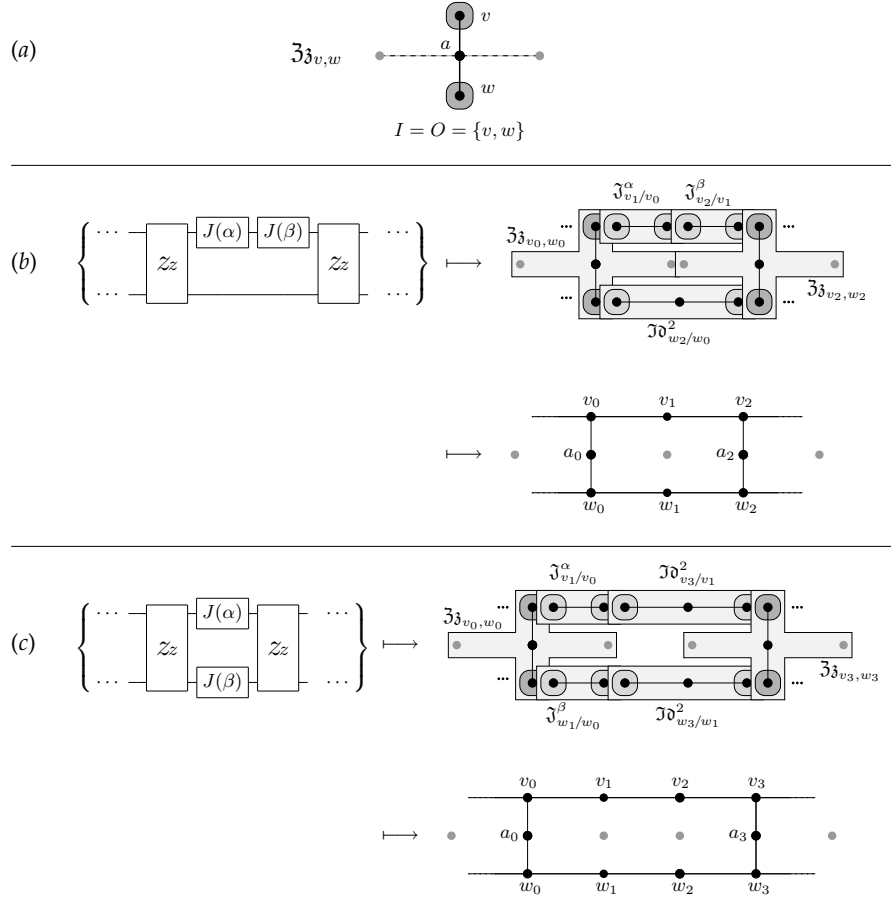


Figure 7: Illustration of the constraints on composition of the pattern \mathfrak{Z}_3 in the setting of an induced subgraph of the grid. Grey dots represent vertices which have been removed from the grid. Composition is illustrated with overlapping tiles, whose shapes represent spatial constraints imposed by the operations.

- (a) The geometry itself, together with the “forbidden” neighbors of the mediating qubit a . (Broken lines between the dots represent the edges removed as a result of removing these vertices.)
- (b) A composition of \mathfrak{Z}_3 procedures with \mathfrak{J}^θ procedures, corresponding to a circuit in which two Zz gates are separated by two $J(\theta)$ operations. The bottom row of the corresponding 1WQC procedure is padded with a \mathfrak{J}^2 procedure to synchronize the lengths of the rows. In this case, a single vertex is a common forbidden neighbor of each mediating qubit.
- (c) A composition of \mathfrak{Z}_3 procedures with \mathfrak{J}^θ procedures, corresponding to a circuit in which two Zz gates are separated by one $J(\theta)$ operation on either wire. The forbidden neighbors of the mediating qubits force both rows to be padded with identity operations.

convert the resulting circuit into stable-index notation: we use an algorithm such as that of Section A.2.1 to do so for the DKP construction; for the simplified RBB construction, we may use instead the procedure of Section A.2.2 to impose additional topological constraints on the resulting circuit. We call the resulting stable-index expression for the circuit C'' .

3. Translate individual operations into the 1WQC model.

In C'' , note that each single-qubit $J(\theta)$ operation deprecates one index and advances another, and that all of the two-qubit $\wedge Z$ or $\mathcal{Z}z$ operations act “stably” on their indices. We may then define a mapping Φ from stable-index expressions of operations in the gate-set $\mathcal{B}_{\text{DKP}} \cup \mathcal{B}_{\text{RBB}}$ (and introduction of fresh qubits in the $|+\rangle$ state) to 1WQC procedures involving \mathfrak{J}^θ , E , and \mathfrak{Z}_3 operations as follows:

$$\Phi\left(|+\rangle\left[\begin{smallmatrix} v \\ \end{smallmatrix}\right]\right) = \mathsf{N}_v ; \quad (2.52a)$$

$$\Phi\left(J(\theta)\left[\begin{smallmatrix} w \\ v \end{smallmatrix}\right]\right) = \mathfrak{J}_{w/v}^\theta ; \quad (2.52b)$$

$$\Phi\left(\wedge Z[v, w]\right) = \mathsf{E}_{vw} ; \quad (2.52c)$$

$$\Phi\left(\mathcal{Z}z[v, w]\right) = \mathfrak{Z}_{v, w} ; \quad (2.52d)$$

where we identify the labels of indices of the stable-index tensor expression with labels for qubits in the 1WQC procedure. We then extend this map Φ homomorphically to products of such terms.

4. Obtain a normal form for the resulting 1WQC procedure.

We standardize (Definition 2.5) the 1WQC procedure $\Phi(C'')$, and then apply Pauli simplifications and signal shifting to obtain a procedure in normal form (Definition 2.6).

By construction of C'' in the above procedure, each index is advanced at most once and deprecated at most once; thus, each qubit in $\Phi(C'')$ is produced in the output of an operation (or “allocated”) at most once, and removed from the input of an operation (or “discarded”) at most once. Furthermore, by order of the terms in C'' , the operation which allocates a qubit v is the first which acts on v , and the operation which discards v is the last acting on v . The resulting 1WQC procedure is therefore well-formed.

By the analysis of Section 2.3.1, we may easily show in either case that $\Phi(C'')$ is a CPTP map which performs the same unitary transformation as described by C'' , modulo an identification of each input qubit with a corresponding output qubit. Then the final procedure produced performs the same transformation as the circuit C . We refer to this procedure as the *simplified RBB construction* for 1WQC procedures when we require that C be linear nearest-neighbor and when we perform the substitution of (2.50), and the *DKP construction* for 1WQC procedures when we do not impose the added constraint or perform the substitution. (For the sake of brevity, we will often refer to this construction simply as “the RBB construction”, as we do not consider any constructions using the special-purpose 1WQC procedures in Section IV of [2].)

Run-time complexity. Both the DKP construction and the (simplified) RBB construction can be performed efficiently in the size of the input circuit. In both cases, let k be the number of qubits which C acts upon, N be the number of one-qubit gates, and M be the number of two-qubit gates: if we consider only those circuits which act non-trivially on each qubit, we may assume that $k \leq N + M$ for simplicity. We may then show that the run-time complexity of the DKP construction is $O((N + k)M)$, and of the RBB construction is $O(NM(N + M))$, in each case bounded by the complexity of obtaining a 1WQC procedure in normal form. (The difference in complexity between the two constructions may be traced to the difference in the procedures to obtain the stable-index expression C'' for each, and hence to the single-qubit gates which are added in the RBB construction in order to impose the desired topological constraints on the indices of the stable-index expression.) An explicit analysis of the complexity of these constructions is presented in Section B.2.

2.3.5 Partial constructions

As a final remark on constructions of 1WQC procedures, it will be useful to describe partial versions of the DKP and (simplified) RBB constructions. We define the *DKP construction without normalization*^P and the *RBB construction without normalization* to be, simply, the 1WQC procedure constructions which result from omitting the Pauli simplifications and signal shifting stages from the phase of obtaining a 1WQC procedure in normal form. A procedure produced by such a partial procedure may then indicate sign dependencies for a measurements whose default angle is an integral multiple of $\pi/2$, and they may have signal-shift operators immediately following measurements.

In a practical setting, provided sufficiently fast control of the classical memory storing the measurement results $s[v]$ for all qubits v in a 1WQC procedure, it is likely that a more useful procedure would be an intermediate construction to the constructions with or without normalization, in which one performs Pauli simplifications but leaves the shift operations in place. The purpose of identifying the partial constructions without normalization above is to obtain a simplified analysis of measurement dependencies, which may be used as a starting point for the analysis of the “complete” constructions.

3 Semantic maps

We may now consider the roles of the DKP and (simplified) RBB constructions as defining representations of unitary circuits in the 1WQC model. Both constructions make use of a simple process of translating unitary circuits from a gate set including $J(\theta)$ gates and a single two-qubit diagonal operation into corresponding 1WQC procedures: this will induce a graph structure which is dominated by vertex-disjoint paths, with supplemental vertices and edges linking pairs of paths. After standardizing the 1WQC procedures in the course of these constructions, the additional structure which might arise from measurement dependencies or the grouping of operations is lost or

^PThis construction is referred to in [9] as the “simplified” DKP construction; we use different terminology here in order to be more uniform with our description of the RBB construction.

obfuscated; preparation maps and entangling operations become dissociated from individual measurement operations, and corrections will in some cases be absent entirely after absorbing them into measurements and performing Pauli simplifications.

Given a 1WQC procedure, in which no additional information is encoded *e.g.* in the labels for the qubits, we may ask whether we can efficiently identify whether a 1WQC procedure is one that may be produced by the DKP or the RBB constructions. In the case where the input and output subsystems of the procedure are of the same size, we answer this question in the affirmative. We do so by considering the combinatorial structures which are present in the geometries (Definition 2.7) of the 1WQC procedures which result from these constructions. We then define a map \mathcal{S} from 1WQC procedures to circuits in the $\{H, T, \wedge Z\}$ model which serves as a semantic map (in the sense of Definition 1.1), for the representation maps \mathcal{R}_{DKP} and \mathcal{R}_{RBB} corresponding to both the DKP and simplified RBB constructions restricted to input circuits which are unitary bijections.

3.1 Measurement dependencies arising in the constructions

The DKP and RBB constructions produce 1WQC procedures from circuits by transforming gates such as $J(\theta)$, $\wedge Z$, and $\mathcal{Z}z$ into simple 1WQC subroutines, via the mapping Φ defined in (2.52). It is instructive to consider the structures described by the sign- and bit-dependencies which arise from composing such 1WQC subroutines, using the partial constructions without normalization described in Section 2.3.5.

3.1.1 Measurement dependencies and index successor functions

For a unitary circuit C consisting of gates from the set $\mathcal{B}_{\text{DKP}} = \{J(\theta), \wedge Z\}_{\theta \in \frac{\pi}{4}\mathbb{Z}}$, consider the last gate $J(\theta)$ performed in the circuit. (If there is more than one such gate which may be performed in parallel, we may choose an arbitrary one.) We may decompose C into circuits $C_\star \tilde{C}$, where C_\star consists of this final $J(\theta)$ gate acting on some qubit u , and any controlled- Z operations acting on u which follow it; \tilde{C} consists of the remaining operations of C . We may refer to C_\star as a *star circuit*^a. Consider the translation of a star circuit C_\star into the 1WQC model via the map Φ defined in (2.52), and standardizing the resulting procedure (Definition 2.5). Writing C_\star in stable-index tensor notation (Section 2.1.3), and letting W be the set of qubits $w \neq u$ on which C_\star acts, we have

$$\begin{aligned} \Phi(C_\star) &= \Phi \left(\left(\prod_w \wedge Z_{[w,v]} \right) J(\theta)_{[u]} \right) \\ &= \left(\prod_{w \in W} E_{wv} \right) \mathfrak{J}_{v/u}^\theta \end{aligned}$$

^aThese star circuits are similar to, but have a different orientation to, the circuits described in Section III B of [12]. This different choice of construction will facilitate the analysis of combinatorial structures later in the article: these are used in much the same way as the circuits arising from “star patterns” do in [12].

$$\begin{aligned}
&= \left(\prod_{w \in W} E_{wv} \right) X_v^{s[u]} M_u^{-\theta} E_{uv} N_v \\
&\cong \left(\prod_{w \in W} Z_w^{s[u]} \right) X_v^{s[u]} M_u^{-\theta} \left(\prod_{w \in W} E_{wv} \right) E_{uv} N_v, \quad (3.1)
\end{aligned}$$

where on the last line we commute the entangling operations to the right. In the final 1WQC procedure, the qubit u is distinguished as the only qubit measured, and v is the only qubit prepared; v is also the only qubit subject to an $X_v^{s[u]}$ operation, with every other qubit (aside from u) being subject to a $Z_w^{s[u]}$ operation. The fact that u and v are measured and prepared qubits (respectively) in $\Phi(C_*)$ above is a direct result of the fact that they are deprecated and advanced indices (respectively) in the stable-index expression given for C_* . In particular we have $v = f(u)$, where f is the index successor function (Definition 2.3) of C_* ; thus, the corrections may be described in terms of the index successor function of the original circuit, and the entangling operations.

We may make similar observations for the (simplified) RBB construction as we have for the DKP construction above. Motivated by the realization of the operator Zz_{vw} in the 1WQC procedure $\mathfrak{Z}_{v,w}$, we may write

$$Zz[v,w] \propto P\left[\frac{a}{a}\right] \wedge Z[v,a] \wedge Z[w,a] |+\rangle\left[\frac{a}{a}\right], \quad (3.2)$$

where we define the operator $P = \langle +_{\pi/2} |$, and for a constant proportionality factor (as illustrated in (B.4) in Section B.1.1). In the stable-index expression on the right, the index successor function f remains the same (as the indices a are advanced without corresponding to any deprecated index, and vice versa), but the interaction graph G is transformed essentially by subdividing every edge arising from a Zz gate by introducing a new vertex in the middle (representing the mediating qubit with the index a). Then, consider the translation of $Zz_{v,w}$ to a 1WQC procedure via Φ ,

$$\Phi\left(Zz(\theta)[u,v]\right) = Z_u^{s[a]} Z_v^{s[a]} M_a^{\pi/2} E_{ua} E_{va} N_a : \quad (3.3)$$

if we extend the successor function f to a function \tilde{f} for which we define $\tilde{f}(a) = a$, while we have no $X_w^{s[a]}$ operations, we have a $Z_w^{s[a]}$ for every qubit w adjacent to $a = \tilde{f}(a)$ in the entanglement graph.

In each case, either the index successor function (or a modest extension of it) for a stable-index description of a simple unitary circuit can be used to characterize the corrections performed in a corresponding 1WQC procedure. We may then consider the result of translating a composition of such simple circuits,

$$\Phi(C) = \Phi\left(C_\star^{(\ell)} \dots C_\star^{(1)}\right) = \Phi\left(C_\star^{(\ell)}\right) \circ \dots \circ \Phi\left(C_\star^{(1)}\right), \quad (3.4)$$

where each $C_\star^{(j)}$ is a star circuit as above, extending the definition to also include individual Zz operations as described in (3.2). Commuting all of the preparation and

entangling operations to the right, we obtain the same dependency of the corrections on the successor function (extended for each $\mathcal{Z}\mathcal{Z}$ gate as described above): we obtain $Z_w^{s[u]}$ corrections for each qubit in the interaction graph which is adjacent to $\tilde{f}(u)$, either due to the commutation of entangling maps past $X_{f(u)}^{s[u]}$ operations or from the $Z_w^{s[u]}$ operations arising from $\mathfrak{Z}\mathfrak{Z}$ procedures.

3.1.2 Efficiently verifying consistency of measurement dependencies

Based on the descriptions of the dependencies arising in the DKP and RBB constructions without normalization, we may characterize the dependencies of operations which arise in the two constructions *with* normalization, and show that verifying that such dependencies hold in a 1WQC procedure \mathfrak{P} can be used as a subroutine to certify that \mathfrak{P} performs a unitary transformation.

Following the remarks of the preceding sections about the constructions without normalization, we characterize the dependencies arising from the DKP and RBB constructions as follows.

Definition 3.1. For a stable-index representation C for a circuit the gate set $\mathcal{B}_{\text{DKP}} \cup \mathcal{B}_{\text{RBB}}$, let \tilde{C} be the stable-index expression obtained by performing the substitution of (3.2) for each $\mathcal{Z}\mathcal{Z}$ operation in C . Let f be the index successor function of \tilde{C} , as defined in Definition 2.3. The *extended successor function* of C is then the function

$$\tilde{f}(v) = \left\{ \begin{array}{ll} f(v), & \text{if } v \text{ an index of } C \\ v, & \text{if } v \text{ is an index deprecated in } \tilde{C} \text{ by a } P \text{ operator} \end{array} \right\}, \quad (3.5)$$

whose domain is the set of indices of \tilde{C} which are deprecated.

Theorem 3.1. Let $\mathfrak{P}_0 = \Phi(C)$, for a unitary circuit C over either the gate set $\mathcal{B}_{\text{DKP}} \cup \mathcal{B}_{\text{RBB}}$, and let \mathfrak{P} be the result of normalizing \mathfrak{P}_0 . Let (G, I, O) be the geometry underlying \mathfrak{P}_0 , and let f be the extended index successor function of C . For the sake of brevity, define $A_X \subseteq O^c$ as the set of qubits to be measured with default angles 0 or π in \mathfrak{P}_0 , and $A_Y \subseteq O^c$ the set of qubits to be measured with default angles $\pm\pi/2$ in \mathfrak{P}_0 . We define square boolean matrices $F, T : V(G) \times V(G) \longrightarrow \mathbb{Z}_2$ in terms of their coefficients, as follows:^r

$$F_{vw} = \left\{ \begin{array}{ll} 1, & \text{if } v \in O^c \text{ and } w = f(v) \notin A_X \cup A_Y \\ 0, & \text{otherwise} \end{array} \right\}, \quad (3.6a)$$

$$T_{vw} = \left\{ \begin{array}{ll} 1, & \text{if } w \neq v \in O^c \text{ and } w \sim f(v) \\ 1, & \text{if } v, w \in O^c \text{ and } w = f(v) \in A_Y \\ 0, & \text{otherwise} \end{array} \right\}. \quad (3.6b)$$

Then $(\mathbb{1} - T)$ is invertible modulo 2, and the dependencies in \mathfrak{P} between qubits $v, w \in V(G)$ may be characterized in terms of F and T as follows:

^rThese operators are the transposes of the operators described in [9], corresponding to changes in the representation in order to simplify the description of the effect of signal shifting; the statement of the result is also changed accordingly.

- w has a sign-dependency on v if and only if $[(\mathbb{1} - T)^{-1}F]_{vw} = 1$, for $w \in O^c$;
- there is an X_w operation conditioned on $s[v]$ if and only if $[(\mathbb{1} - T)^{-1}F]_{vw} = 1$, and a Z_w operation conditioned on $s[v]$ if and only if $[(\mathbb{1} - T)^{-1}T]_{vw} = 1$, for $w \in O$.

Proof. By construction, for qubits $u, v \in V(G)$, \mathfrak{P}_0 contains a $Z_v^{s[u]}$ operation if and only if $v \sim f(u) = u$; as $Z_v^{s[u]}$ operations commute with entangling maps, we obtain no further corrections depending on u if we commute the entangling maps of \mathfrak{P}_0 to the right. Similarly, \mathfrak{P}_0 contains an $X_v^{s[u]}$ operation if and only if $v = f(u) \neq u$, which must arise from a $\mathfrak{J}_{v/u}^\theta$ procedure. Any operation on v in \mathfrak{P}_0 which is not a part of the $\mathfrak{J}_{v/u}^\theta$ procedure must come after it, every entangling map acting on v must occur after the $X_v^{s[u]}$ operation except for the E_{uv} map occurring as a part of $\mathfrak{J}_{v/u}^\theta$.

Commuting all of the entangling maps in \mathfrak{P}_0 to the right, we then induce $Z_w^{s[u]}$ operations for all $w \sim v$ such that $w \neq u$, as described in (2.23c). Thus, commuting all preparation and entangling maps in \mathfrak{P}_0 to the right yields an equivalent procedure \mathfrak{P}_1 , in which for there is an $X_w^{s[v]}$ operation for every $w = f(v) \neq v$, and a $Z_w^{s[v]}$ operation for every $w \sim f(v) \neq v$, for every qubit $v \in O^c$. That is, we have

$$\mathfrak{P}_1 = \left[\prod_{u \in O^c} \leq \left(\prod_{\substack{w \sim f(u) \\ w \neq u}} Z_w^{s[u]} \right) \left(\prod_{\substack{v = f(u) \\ v \neq u}} X_v^{s[u]} \right) M_u^{\theta_u} \right] E_G N_{I^c} \quad (3.7)$$

for some linear order \leq satisfying the conditions of (2.14) for the extended successor function f .

Commuting correction operations in \mathfrak{P}_1 to the left, we obtain another procedure \mathfrak{P}_2 , in which the same corrections hold as above for $w \in O$. For $w \in O^c$, there is instead a sign-dependency on v if and only if $w = f(v) \neq v$, and a bit-dependency on v if and only if $w \neq v$ and $w \sim f(v)$. Note that the procedure of producing \mathfrak{P}_2 from \mathfrak{P}_0 is precisely that of standardizing \mathfrak{P}_0 : then \mathfrak{P} may be obtained by normalizing \mathfrak{P}_2 .

Consider how the process of normalization effects dependencies of qubits w on a particular qubit v .

- The effect of Pauli simplifications are to remove sign-dependencies from qubits in A_X according to (2.34f), and to change them to bit-dependencies for qubits in A_Y according to (2.34g). As signal shifting does not introduce sign-dependencies for measurements where none previously exist, the measurement of a qubit in $A_X \cup A_Y$ will not have any dependencies on previous measurements in a procedure in normal form.

Applying Pauli simplifications to \mathfrak{P}_2 will yield a 1WQC procedure \mathfrak{P}_3 : then \mathfrak{P}_3 will have no sign-dependencies for qubits $v \in A_X \cup A_Y$, and will have signal shift operators $S_{s[v]}^{s[u]}$ for any qubits $v = f(u) \in A_Y$, in addition to the shift operators in \mathfrak{P}_2 . We may describe the sign dependencies which remain in \mathfrak{P}_3 by a square matrix over $V(G)$, with a 1 in the row v and column w for $v, w \in V(G)$ when w has a sign-dependency on v . Accounting for the removals of sign dependencies of \mathfrak{P}_2 for qubits $w \in A_X \cup A_Y$, the procedure \mathfrak{P}_3 then

has a sign-dependency of w on v when $w = f(v) \notin A_X \cup A_Y$. The matrix described in this manner is then the matrix F in (3.6a).

For the sake of simplicity of discussion, we will assume that the shift operators are not accumulated with any pre-existing shift operators using the relation of (2.35). However, it is important to note that the shift operators arising from Pauli simplifications will not cancel any operators existing originally in \mathfrak{P}_2 : if $w = f(v)$, it follows that w is not adjacent to $f(v)$, as G contains no loops by construction.

- (ii) The effect of signal shifting is to propagate a bit-dependency of each qubit w on v (represented by a $S_{s[w]}^{s[v]}$ operation just after the measurement of w) to the operations which in turn depend on the value of $s[w]$, according to (2.36).

We may describe the effect of signal shifting in terms of walks in a directed graph D of dependencies, where we have arcs $w \rightarrow v$ when there is a shift operator $S_{s[w]}^{s[v]}$. An operation depending on w may be transformed into one depending on w and all qubits w for which D has an arc $w \rightarrow v$ by shifting of operators $S_{s[w]}^{s[v]}$; in turn, the resulting operation may be transformed into one depending also on qubits u for which D has arcs $v \rightarrow u$, ranging over the qubits v of the previous step; and so on. The effect is then of transforming bit-dependencies corresponding to arcs in this digraph D (arising from signal shifts) into measurement and correction dependencies corresponding to directed walks in D .

We may represent this digraph as an adjacency matrix over $V(G)$, with a 1 in the row v and column w for $v, w \in V(G)$ when \mathfrak{P}_3 contains a shift operator $S_{s[w]}^{s[v]}$, and 0 if not. Considering the shift operations in \mathfrak{P}_3 which are also present in \mathfrak{P}_2 , there will be such a shift operator whenever $w \neq v$ and $w \sim f(v)$; and as a result of Pauli simplifications, there will be such a shift operator when $w = f(v)$ and $w \in A_Y$. As there are no other sources of shift operators, these are the only positions in which the adjacency matrix will be non-zero. The adjacency matrix described is then the matrix T in (3.6b).

As the ordering \leq (used in (3.7) to fix the order of operations) is a linear order, the digraph D described by the shift operations is acyclic: for any standard basis vector $\hat{e}_w \in \mathbb{Z}_2^{V(G)}$, the column-vector $T\hat{e}_w$ is supported only on indices $v \leq w$. Then we have $T^r = 0$ for some $r \leq |V(G)|$, so that T is nilpotent (and in particular has no +1 eigenvectors). Consequently, the operator $\mathbb{1} - T$ is invertible. We may then express the cumulation of dependencies for each operation in \mathfrak{P}_3 due signal shifting as follows. For two qubits v and w , the number of walks in D from w to v of a fixed length $\ell \geq 0$ is given by $\hat{e}_v^\top T^\ell \hat{e}_w$: the total number of walks of any length from w to v in D is then

$$\hat{e}_v^\top \left(\sum_{\ell=0}^{\infty} T^\ell \right) \hat{e}_w = \hat{e}_v^\top (\mathbb{1} - T)^{-1} \hat{e}_w. \quad (3.8)$$

If we represent the dependencies of operations in terms of boolean column vectors, we may then represent the effect of signal shifting by multiplying a vector of dependencies by $(\mathbb{1} - T)^{-1}$ computed modulo 2.

To determine the classical dependencies of each operation in $\bar{\mathfrak{P}}$, it suffices then to identify the dependencies of the same operations in \mathfrak{P}_3 , represent these as a vector $\mathbf{d} \in \mathbb{Z}_2^{V(G)}$, and compute $(\mathbb{1} - T)^{-1} \mathbf{d}$. We may characterize the classically controllable operations and their dependencies in \mathfrak{P}_3 as follows:

- As we have noted above, the sign-dependencies for qubits $w \in O^c$ are represented by the coefficients of the matrix F , where $F_{vw} = 1$ if and only if w has a sign dependency on v in \mathfrak{P}_3 . Then w has a sign dependency on v in $\bar{\mathfrak{P}}$ if and only if $[(\mathbb{1} - T)^{-1} F]_{vw} = 1$.
- For $w \in O$, there is no measurement into which corrections on w may be absorbed, and so the corrections in \mathfrak{P}_3 are the same as in \mathfrak{P}_1 : these are $X_w^{s[v]}$ operations when $w = f(v)$, and $Z_w^{s[v]}$ operations when $w \sim f(v)$, where in either case we require $w \neq v$ as well. As O is disjoint from both A_X and A_Y by definition, we may then note that for $w \in O$ there is an $X_w^{s[v]}$ operation in \mathfrak{P}_3 if and only if $F_{vw} = 1$, and a $Z_w^{s[v]}$ operation if and only if $T_{vw} = 1$. It follows that after signal shifting, the resulting X_w^β operation depends on v if and only if $[(\mathbb{1} - T)^{-1} F]_{vw} = 1$, and the resulting Z_w^β operation depends on v if and only if $[(\mathbb{1} - T)^{-1} T]_{vw} = 1$. ■

Theorem 3.1 thus characterizes the dependencies which arise in the DKP and RBB constructions, in terms of an (extended) index successor function for a circuit C over the gate-set $\mathcal{B}_{\text{DKP}} \cup \mathcal{B}_{\text{RBB}}$.

Algorithm 1 describes an algorithm which, provided a 1WQC measurement pattern \mathfrak{P} in normal form and a candidate for the extended successor function f for an originating circuit, tests if the dependencies of \mathfrak{P} are consistent with the dependency conditions described in Theorem 3.1.

If we are presented with a 1WQC procedure \mathfrak{P} in normal form which is *not* known to be a result of the DKP or RBB constructions, we may only verify whether these conditions hold relative to some “candidate” successor function f . How such a function f may be constructed for a 1WQC procedure \mathfrak{P} is the subject of Section 3.2.

Run-time analysis of Algorithm 1. We bound the run-time of Algorithm 1 as follows. In the following, we let N be the number of operations in \mathfrak{P} , and for the geometry (G, I, O) underlying \mathfrak{P} , we let $n = |V(G)|$, $k = |O|$, and $m = |E(G)|$. We assume that f can be evaluated in constant time, using an array structure. We assume that the dependencies of operations Op^β (for sign-dependencies or other classical control expressions) are represented as indicator vectors $\mathbf{d} \in \mathbb{Z}_2^{V(G)}$ representing the presence or absence of the term $s[v]$ in β by $d_v = 1$ (with $d_v = 0$ otherwise).

By definition, \mathfrak{P} will contain m entangling operations, $n - k$ measurement operations, and at most $2k$ correction operations; then, the length of \mathfrak{P} is $N \in O(m + n) \subseteq O(n^2)$. Rather than compute matrix products involving $(\mathbb{1} - T)^{-1}$, we may verify whether $(\mathbb{1} - T)\mathbf{d} = \mathbf{v}$ for $\mathbf{v} = F\hat{\mathbf{e}}_w$ or $\mathbf{v} = T\hat{\mathbf{e}}_w$ as appropriate. The coefficients of $(\mathbb{1} - T)\mathbf{d}$ may be computed by iterating over the neighbors of $f(w)$ for $w \in {}^{(\cdot)}\mathbf{d}$: as $T_{vw} = 1$ only for w adjacent to $f(v)$ (if $v \neq w$) or for $w = f(v)$, there are at most two non-zero coefficients in T for each edge in G , so that this iteration requires time $O(m)$ for each vector \mathbf{d} . Similarly, the time required to compute column-vectors

Input : \mathfrak{P} , a 1WQC procedure with geometry (G, I, O) ;
 $f : O^c \rightarrow I^c$, a candidate successor function for \mathfrak{P} .
Output: A boolean result indicating if the dependency relations in \mathfrak{P} are consistent with Theorem 3.1, relative to the function f .

```

1 procedure TestDependencies( $\mathfrak{P}, f$ ) :
2   Let  $T$  be a matrix depending on  $f$  as in (3.6a).
3   Let  $F$  be a matrix depending on  $f$  as in (3.6b).
4   Let  $\tilde{T} \leftarrow (\mathbb{1}_n - T)$ .
5   For each measurement or correction operation  $\text{Op}$  in  $\mathfrak{P}$ , do:
6     If  $\text{Op} = M_w^{\theta;\beta}$  for some  $w, \theta, \beta$ , then
7       Let  $\mathbf{d} \in \mathbb{Z}_2^{V(G)}$  be the indicator vector for  $\beta$ .
8       If  $\tilde{T}\mathbf{d} \neq F\hat{\mathbf{e}}_w$ , then return false.
9     else, if  $\text{Op} = X_w^\beta$  or  $\text{Op} = Z_w^\beta$  for some  $w, \beta$ , then
10      Let  $\mathbf{d} \in \mathbb{Z}_2^{V(G)}$  be the indicator vector for  $\beta$ .
11      If  $(\text{Op} = X_w^\beta \text{ and } \tilde{T}\mathbf{d} \neq F\hat{\mathbf{e}}_w) \text{ or } (\text{Op} = Z_w^\beta \text{ and } \tilde{T}\mathbf{d} \neq T\hat{\mathbf{e}}_w)$ 
12        then return false.
13    endif
14  endfor
15  return true.

```

ALGORITHM 1. An algorithm to test whether the dependencies of a 1WQC procedure is consistent with those resulting from the DKP and simplified RBB constructions, relative to a candidate for the (extended) index successor function for the original circuit.

$F\hat{\mathbf{e}}_w$ and $T\hat{\mathbf{e}}_w$ for each w is $O(m)$. As there are $O(n)$ measurement and correction operations, the cumulative time to perform these matrix computations is $O(nm)$; as the remaining operations in \mathfrak{P} require no verification, this is the total time complexity of the **for** loop. The execution time of Algorithm 1 is then $O(nm)$.

3.2 Conditions on candidate successor functions

In the preceding section, we characterized the dependencies in the DKP and RBB constructions in terms of an extended index successor function, from a stable-index representation of the original circuits. This leaves open the question of how such a successor function may be obtained. We examine this problem using the observations made above about the dependencies arising from circuit decompositions without normalization. We show that these dependencies can be captured in each case by a modification of the *flow* conditions formulated by Danos and Kashefi [12]. This modification will provide the “candidate successor functions” which we require for the verification procedure of Algorithm 1, and which lead to important structural constraints on the geometries (G, I, O) arising from the DKP and RBB constructions.

3.2.1 Measurement ordering from index successor functions

As we described in Section 2.3.2, there is a correspondence between the “deprecation order” of indices as described in (2.14), for a stable-index expression for a circuit C over $\mathcal{B}_{\text{DKP}} \cup \mathcal{B}_{\text{RBB}}$, and the order of operations in a well-formed 1WQC procedure $\Phi(C)$. This correspondence may be attributed to a sense in which measuring qubits in a 1WQC procedure simulates the summation over indices in a tensor expression, and thus to evaluating a unitary circuit applied to a state as a sum over computational paths.

When applied to a completely specified pure state, every deprecated index in a stable-index expression becomes bound, in which one evaluates the result by summing over all deprecated indices. When these indices are represented by qubits in a 1WQC procedure, one may simulate this summation by measurement of qubits v with respect to the $|+\theta\rangle, |-\theta\rangle$ basis.

- Consider a measurement arising from a \mathfrak{J}^θ procedure, corresponding to summing over the deprecated index of a $J(\theta)$ operation. If the $|+\theta\rangle$ state is measured (*i.e.* if $s[u] = 0$), the resulting transformation corresponds to summing over the $|0\rangle$ and $|1\rangle$ components, weighted with the relative amplitudes phases as those associated with the $J(\theta)$ operation which deprecates the corresponding index.
- For measurements arising from a $\mathfrak{J}_{v,w}$ procedure, we measure a mediating qubit a in a fixed basis $|\pm\pi/2\rangle$. Again, a measurement result of $|+\pi/2\rangle$ corresponds to summing over an auxiliary index a in a stable-index expression: the relative phases of the $|0\rangle$ and $|1\rangle$ components correspond to those of the projection operator P .

In either case, if the measurement yields the $|-\theta\rangle$ result instead, the corrections and measurement adaptations are chosen precisely to effect the same transformation which would have occurred for the $|+\theta\rangle$ result, by influences on the measurement angles due to sign-dependencies or the measurement results due to bit-dependencies on v .^s

Thus, we may interpret the sign- and bit-dependencies in such 1WQC computations as potential “influences” in evaluating the summation of deprecated indices. By construction, the measurements are chosen in such a way that the tensor which is produced at the output is related to the tensor at the input by a unitary circuit: the amplitudes arising from each measurement are a function of at most two boolean indices, where the two-index dependencies are signs encoded by the entanglement graph, and whose single-index dependencies are given by the choice of measurement basis. This may be regarded as indicating that a 1WQC procedure where such an ordering is possible has a “circuit-like structure”.

A measurement order which corresponds to a “deprecation order” for indices in a circuit C as in (2.14) is governed by the index successor function of C . An extended index successor function as defined in Definition 3.1 has the effect of imposing constraints on the deprecation of the mediating indices a arising from the substitution of

^sBit-dependencies, as represented by shift operators $S_{s[v]}^*$ immediately after measurements, may be informally interpreted as flipping the bit $s[v]$ to produce “the result which would have been measured if an appropriate Z correction had been performed prior to measurement”. As the measurement results are often maximally random whether or not such corrections are performed, this is a somewhat counterfactual interpretation, but one which could perhaps be given an ontological foundation in terms of hidden variables.

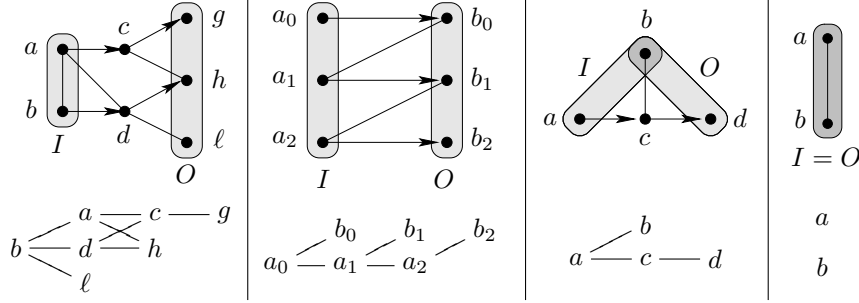


Figure 8: Examples of geometries with flows. Arrows indicate the action of a function $f : O^c \rightarrow I^c$ along otherwise undirected edges. Corresponding partial orders \preceq for each example are given by Hasse diagrams below the graphs (with minimal elements on the left and maximal elements on the right). In the right-most example, the two vertices a and b are incomparable, *i.e.* there is no order relation between them.

(3.2) for \mathcal{Zz} gates, which may be interpreted as creating a well-defined notion of when the \mathcal{Zz} operation is performed. By construction, 1WQC procedures arising from the DKP and RBB constructions do admit extended successor functions: this can be used to define the domain of a semantic map for these constructions.

3.2.2 Flow conditions

The ordering in (2.14) was first formulated by Danos and Kashefi [12], for qubits in the 1WQC model rather than tensor indices, to describe dependencies arising in the DKP construction without normalization. We may formulate these properties in terms of the geometry of a 1WQC procedure without reference to the DKP construction as follows:

Definition 3.2. For a geometry (G, I, O) , let $u \sim v$ denote the adjacency relation in G . A *flow* is a pair (f, \preceq) consisting of an injective function $f : O^c \rightarrow I^c$, and a partial order \preceq on $V(G)$, such that the conditions

$$v \sim f(v) , \quad (3.9a)$$

$$v \preceq f(v) , \text{ and} \quad (3.9b)$$

$$w \sim f(v) \implies v \preceq w \quad (3.9c)$$

hold for all $v \in O^c$ and $w \in V(G)$.

Examples of geometries with and without flows are illustrated in Figure 8 and Figure 9, respectively. The function f and the partial order \preceq capture the essential structure of the dependencies in the DKP construction without normalization: f represents the mapping of qubits to their successors (corresponding to an index successor function for

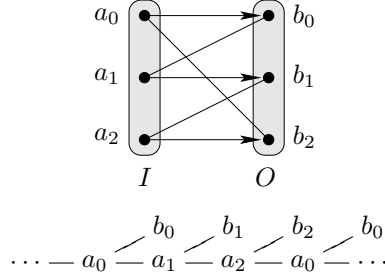


Figure 9: Example of a geometry with no flow (*c.f.* the second geometry from the left in Figure 8). Arrows indicate the action of an injective function $f : O^c \rightarrow I^c$ along otherwise undirected edges. Also given is a reflexive and transitive binary relation \preceq which satisfies conditions (3.9b) and (3.9c) for this function f , but which is not antisymmetric.

a circuit), and the partial order \preceq represents a suggested order in which the qubits may be measured (or the tensor indices deprecated).

Danos and Kashefi show that any geometry which has a flow underlies some 1WQC measurement pattern which can be obtained by the DKP construction,^t and which therefore performs a unitary transformation:

Lemma 3.2 ([12], Theorem 1). *Suppose (f, \preceq) is a flow for (G, I, O) . Let $v \sim w$ denote the adjacency relation of G : then for any linear order \leq extending \preceq , the measurement procedure*

$$\left[\prod_{u \in O^c} \left(\prod_{\substack{w \sim f(u) \\ w \neq u}} Z_w^{s[u]} \right) X_{f(u)}^{s[u]} M_u^{\theta_u} \right] E_G N_{I^c} \quad (3.10)$$

performs a unitary transformation.

The proof presented in [12] is essentially by showing that applying the map Φ defined in (2.52) to a suitably constructed circuit yields the 1WQC procedure above. If we can construct a flow for a geometry (G, I, O) , we may then use the same circuit construction as in [12] to construct a candidate circuit whose representation as a 1WQC procedure (via the DKP construction) has the same geometry. We may then verify whether the dependencies of a 1WQC procedure are consistent with one arising from the DKP construction, using Algorithm 1.

^tThe converse of this statement does not necessarily hold: by omitting or manipulating the dependencies in a 1WQC procedure, a variant procedure which does *not* perform a unitary may be obtained without changing the underlying geometry.

3.2.3 Extending to modified flows

By considering the dependencies described in the proof of Theorem 3.1 following (3.7), we may formulate a similar combinatorial condition which generalizes flows, and also captures the dependencies arising in the (simplified) RBB construction without normalization:^u

Definition 3.3. Let (G, I, O, M) consist of a geometry (G, I, O) and a subset $M \subseteq O^c$, and let $u \sim v$ denote the adjacency relation in G . A *modified flow* for (G, I, O, M) is an ordered pair (f, \preceq) consisting of an injective function $f : O^c \longrightarrow I^c$, and a partial order \preceq on $V(G)$, such that the conditions

$$f(v) \sim v, \text{ for } v \in M^c; \quad (3.11a)$$

$$\text{either } f(v) = v \text{ or } f(v) \sim v, \text{ for } v \in M; \quad (3.11b)$$

$$v \preceq f(v); \text{ and} \quad (3.11c)$$

$$w \sim f(v) \implies v \preceq w \quad (3.11d)$$

hold for all $v \in O^c$ and $w \in V(G)$.

The function f again corresponds essentially to an (extended) successor function for stable-index expression of a unitary circuit, and \preceq again represents an order of measurement corresponding to an order in which the indices of stable-index expression are deprecated. The set M represents qubits whose measurement angles are $\pi/2$: these are qubits which might be mediating qubits a in the $\mathfrak{J}_{v,w}$ procedures. We may recover the definition for flows in Definition 3.2 if we either set $M = \emptyset$, or simply require that $v \neq f(v)$ for all v .

The following generalization of Theorem 1 of [12] to modified flows provides the basis for the approach we will take to define semantic maps for the DKP and RBB constructions:

Lemma 3.3. Suppose (f, \preceq) is a modified flow for (G, I, O, M) . Let $v \sim w$ denote the adjacency relation of G : then for any linear order \leq extending \preceq , the measurement procedure

$$\left[\prod_{u \in O^c} \leq \left(\prod_{\substack{w \sim f(u) \\ w \neq u}} z_w^{s[u]} \right) \left(\prod_{\substack{v = f(u) \\ v \neq u}} x_v^{s[u]} \right) M_u^{\theta_u} \right] E_G N_{I^c} \quad (3.12)$$

performs a unitary transformation, provided $\theta_u = \pi/2$ for every $u \in M$. Furthermore, for 1WQC procedures \mathfrak{P} arising from the DKP or (simplified) RBB constructions, the geometry (G, I, O) underlying \mathfrak{P} (together with the set M of qubits with measurement angle $\pi/2$) has a modified flow.

^uA similar generalization of the flow conditions was anticipated in [12], but does explicitly specify combinatorial conditions for this generalization.

Proof. We may obtain a procedure as above using a map $\bar{\Phi}$ extending Φ , as follows. Let $\bar{\mathcal{B}} = \{J(\theta), \wedge Z, \mathcal{Z}\mathcal{Z}_d\}_{\theta \in \frac{\pi}{4}\mathbb{Z}, d \geq 2}$, where we define the d qubit operator

$$\mathcal{Z}\mathcal{Z}_d = \exp(-i\pi Z^{\otimes d}/4) = e^{-i\pi \underbrace{[Z \otimes \dots \otimes Z]_{d \text{ times}}}/4}. \quad (3.13)$$

We may define the 1WQC procedure $\mathfrak{Z}\mathfrak{Z}_d$ analogously to $\mathfrak{Z}\mathfrak{Z}$, as follows:

$$\mathfrak{Z}\mathfrak{Z}_{v_1, \dots, v_d}^d = \left(\prod_{j=1}^d Z_{v_j}^{s[a]} \right) M_a^{\pi/2} \left(\prod_{j=1}^d E_{av_j} \right) N_a : \quad (3.14)$$

by an analysis deferred to Section B.1.1 (see also Section 3.8 of [19]), we may show

$$\mathfrak{Z}\mathfrak{Z}_{v_1, \dots, v_d}^d(\rho_S) = [\mathcal{Z}\mathcal{Z}_d]_{v_1, \dots, v_d} \rho_S [\mathcal{Z}\mathcal{Z}_d]_{v_1, \dots, v_d}^\dagger, \quad (3.15)$$

for a system S including the qubits v_j , but excluding a . For a stable-index expression for gates in \mathcal{B} , we may then define:

$$\bar{\Phi}\left(|+\rangle\left[\frac{v}{v}\right]\right) = N_v; \quad (3.16a)$$

$$\bar{\Phi}\left(J(\theta)\left[\frac{w}{v}\right]\right) = \mathfrak{J}_{w/v}^\theta = X_w^{s[v]} M_v^{-\theta} E_{vw} N_w; \quad (3.16b)$$

$$\bar{\Phi}\left(\wedge Z[v, w]\right) = E_{vw}; \quad (3.16c)$$

$$\bar{\Phi}\left(\mathcal{Z}\mathcal{Z}_d[v_1, \dots, v_d]\right) = \mathfrak{Z}\mathfrak{Z}_{v_1, \dots, v_d}^d, \quad (3.16d)$$

extending this to unitary circuits over $\bar{\mathcal{B}}$ in the usual manner. For a star circuit $C_\star = \wedge Z[v, w_1] \dots \wedge Z[v, w_k] J(\theta_u)\left[\frac{v}{u}\right]$ as in (3.1), we then have

$$\bar{\Phi}(C_\star) \cong \left(\prod_{j=1}^n Z_{w_j}^{s[u]} \right) X_v^{s[u]} M_u^{-\theta} E_{G_\star} N_v \quad (3.17)$$

after commuting the entangling maps to the right, where G_\star is a star graph with center v and edges from v to each w_j and to u ; we may define a similar star graph for the entangling maps of $\mathfrak{Z}\mathfrak{Z}_{v_1, \dots, v_d}^d$ so that we may write

$$\bar{\Phi}(\mathcal{Z}\mathcal{Z}_d) \cong \left(\prod_{j=1}^d Z_{w_j}^{s[a]} \right) M_a^{-\theta} E_{G_\star} N_v \quad (3.18)$$

for such a star-graph.

Given a tuple (G, I, O, M) with a modified flow (f, \preceq) , define the circuits C_u for $u \in O^c$ via the following stable-index expressions:

$$C_u = \left(\prod_{\substack{w \sim f(u) \\ w \notin \{u, f^2(u)\}}} \wedge Z[f(u), w] \right) J(\theta_u)\left[\frac{f(u)}{u}\right], \quad \text{for } u \neq f(u); \quad (3.19a)$$

$$C_u = (\mathcal{Z}z_d)_{w_1, \dots, w_d}, \quad \text{for } u = f(u) \text{ with neighbors } w_1, \dots, w_d. \quad (3.19b)$$

We also define a circuit $C_{\tilde{I}}$ by

$$C_{\tilde{I}} = \prod_{\substack{uv \in E(G) \\ u, v \notin \text{img}(f)}} \wedge Z[u, v] : \quad (3.20)$$

we may describe the $\wedge Z$ operations in this circuit by the induced subgraph $G[\tilde{I}]$, where $\tilde{I} = V(G) \setminus \text{img}(f)$. Consider the circuit \mathcal{C} described by the stable-index expression consisting of the concatenation of these circuits. Placing the circuit $C_{\tilde{I}}$ at the beginning, and ordering the remaining circuits C_u in a linear order \leq extending \preceq , we obtain an expression in which operators with index $u \neq f(u)$ occur no later than any operator involving $f(u)$ or $w \sim f(u)$, and any operator with index $u = f(u)$ occurs before the indices $w \sim u$ are deprecated. Thus, applying the homomorphism $\bar{\Phi}$ to each sub-circuit C_u in \mathcal{C} yields a well-formed 1WQC procedure

$$\mathfrak{P} \cong \left[\prod_{u \in O^c} \leq \left(\prod_{\substack{w \sim f(u) \\ w \notin \{u, f^2(u)\}}} Z_w^{s[u]} \right) \left(\prod_{\substack{v=f(u) \\ v \neq u}} X_v^{s[u]} \right) M_u^{\theta_u} E_{G_u} N_{f(u)} \right] E_{G[\tilde{I}]} , \quad (3.21)$$

where G_u is the star graph for each qubit u as described above, and the congruence arises from commutation of preparation maps and entangling maps to the right of each term in the ordered product.

Note that for every edge $vw \in E(G)$, we have either $v, w \notin \text{img}(f)$, or one of the vertices (without loss of generality, v) is of the form $f(u)$ for some $u \in O^c$. In the former case, vw corresponds to a $\wedge Z$ operation in $C_{\tilde{I}}$, and thus to an entangling operation in $\bar{\Phi}(C_{\tilde{I}})$; in the latter case, we have $w \sim f(u)$, in which case vw corresponds to an entangling operation in the star-graph G_u and thus to an entangling operation in $\bar{\Phi}(C_u)$. Conversely, every entangling operation in \mathfrak{P} above corresponds to some edge in G , by construction; the geometry underlying \mathfrak{P} is then (G, I, O) . Note also that the individual operations in \mathfrak{P} and the procedure described in (3.12) are identical, except that corrections of the form $Z_{f^2(u)}^{s[u]}$ do not occur in \mathfrak{P} ; commuting all the preparation and entangling maps to the right of the expression, these are induced by commuting operations $E_{f(u), f^2(u)}$ past the corrections $X_{f(u)}^{s[u]}$. Thus, (3.12) is the result of performing this commutation to \mathfrak{P} ; the 1WQC procedure in (3.12) thus performs the same unitary transformation as \mathcal{C} .

Finally, we may show that every geometry underlying a 1WQC procedure obtained from the DKP or (simplified) RBB constructions has a modified flow. Consider a stable-index expression C for a circuit over the gate-set $\mathcal{B}_{\text{DKP}} \cup \mathcal{B}_{\text{RBB}}$, whose terms of C are ordered according to the order of performance in the circuit. Let f be the extended successor function of C , and \leq be the order in which the indices are deprecated in C . Then f will be a mapping from the deprecated indices to the advanced

indices, and (f, \leq) satisfies (2.14) by construction. If (G, I, O) is the geometry underlying the procedure $\Phi(C)$, and M is the set of qubits with measurement angle $\pi/2$, by construction (f, \leq) will be a modified flow for (G, I, O, M) ; as the DKP and RBB constructions both involve producing such a stable-index expression C , and the process of normalizing a 1WQC procedure leaves the geometry invariant, the result holds. ■

The foregoing proof not only demonstrates that a 1WQC procedure as in (3.12) performs a unitary transformation, but gives a description of a unitary circuit performing the same transformation. This will form the basis of a similar construction in Section 3.4 for constructing circuits without reference to corrections or measurement dependencies.

The proof above also describes a construction of 1WQC procedures based on a gate-set which subsumes $\mathcal{B}_{\text{DKP}} \cup \mathcal{B}_{\text{RBB}}$, containing operations acting on arbitrarily many qubits. The extension to this gate-set $\tilde{\mathcal{B}}$ from \mathcal{B}_{DKP} and \mathcal{B}_{RBB} corresponds to the way in which modified flows extend the pattern of dependencies found in either the DKP or RBB constructions without normalization; the inclusion of the operators \mathcal{Z}_{zz_d} for $d > 2$ is essentially a byproduct of the lack of further constraints on the definition of modified flows.

3.2.4 The natural pre-order, and uniqueness in the case $|I| = |O|$

Following [13], for any geometry (with or without a modified flow) and for any function $f : O^c \longrightarrow I^c$, we may consider an ordering satisfying the conditions of (3.9) or (3.11) as follows:

Definition 3.4. Let (G, I, O) be a geometry and $f : O^c \longrightarrow I^c$. The *natural pre-order* \preceq for f is the reflexive and transitive closure of the conditions

$$v \preceq f(v) \tag{3.22a}$$

$$w \sim f(v) \implies v \preceq w \tag{3.22b}$$

for all $v \in O^c$ and $w \in V(G)$.

Note that the conditions of (3.22) are exactly the conditions (3.11c) and (3.11d) for modified flows. The definition of the natural pre-order then relaxes the condition that \preceq be a partial order, *i.e.* that \preceq is antisymmetric.

If the natural pre-order \preceq happens also to be a partial order, then (f, \preceq) is a modified flow. By the definition of the transitive closure, any pre-order satisfying the conditions of (3.22) must also contain the relations of the natural pre-order \preceq ; then if \preceq is not antisymmetric, there can be no antisymmetric relation \leq satisfying the modified flow conditions with f .

Because of the distinguished nature of the natural pre-order, we may prove a useful uniqueness result for modified flows:

Theorem 3.4. Let (G, I, O) be a geometry and $M \subseteq O^c$. If $|I| = |O|$, then there is at most one function f such that (f, \preceq) is a modified flow for (G, I, O, M) , for \preceq the natural pre-order of f .

Proof. Suppose that $|I| = |O|$. If $I = O$, then a tuple (f, \preceq) can only be a flow if $f(v) = v$ for every $v \in O^c$, as otherwise we obtain cycles $v \prec f(v) \prec \dots \prec f^k(v) = v$ for some v . Then the only function f for which (f, \preceq) may be a modified flow is the identity function on O^c .

Otherwise, suppose that $I \neq O$, and consider two functions $f, g : O^c \longrightarrow I^c$ such that $f(v) = v$ only for $v \in M$, and similarly for g . Note that the natural pre-order \preceq of f can only be anti-symmetric if there are no cycles of the form $v \prec f(v) \prec \dots \prec f^k(v) = v$; thus, we require that f have no such cycles for $k > 1$, and again similarly for g .

Suppose f and g differ, and let $u_0 \in O^c$ be some vertex for which $f(u_0) \neq g(u_0)$. Let $\tilde{g} : I^c \longrightarrow O^c$ be the inverse function of g : we may recursively construct an unbounded sequence of vertices u_j for $j \in \mathbb{N}$ by defining $u_1 = f(u_0)$, and

$$u_{j+1} = \begin{cases} f(u_j), & \text{if } u_j \neq f(u_{j-1}) \\ \tilde{g}(u_j), & \text{if } u_j = f(u_{j-1}) \end{cases} \quad (3.23)$$

for $j \geq 1$. Note that $\text{dom}(\tilde{g}) = I^c = \text{img}(f)$ and $\text{dom}(f) = O^c = \text{img}(\tilde{g})$; then this sequence is well defined for all $j \geq 0$. We may also show that $u_{j+1} \neq u_{j-1}$ for each $j \geq 1$ by induction, as follows. Suppose that $u_{j+1} \neq u_{j-1}$ for some particular $j \geq 1$:

- Suppose that $u_{j+1} = \tilde{g}(u_j) \neq f(u_j)$: it follows that $u_j = f(u_{j-1})$, so that $u_{j+2} = f(u_{j+1}) \neq f(u_{j-1}) = u_j$ by the injectivity of f .
- Suppose that $u_{j+1} = f(u_j) \neq \tilde{g}(u_j)$: it follows that $u_j = \tilde{g}(u_{j-1}) \neq f(u_{j-1})$, in which case we have $u_{j+2} = \tilde{g}(u_{j+1}) \neq \tilde{g}(u_{j-1}) = u_j$, by the injectivity of g .
- Suppose that $u_{j+1} = f(u_j) = \tilde{g}(u_j)$. Because f is injective, we have $u_{j+1} = f(u_j) \neq u_j$ if $u_j \neq u_{j-1}$; and as $u_{j+1} \neq u_{j-1}$, at least one of the inequalities $u_{j+1} \neq u_j$ or $u_j \neq u_{j-1}$ must hold. Thus, we have $u_{j+1} = \tilde{g}(u_j) \neq u_j$ in any case; by the injectivity of \tilde{g} , we then have $u_{j+2} = \tilde{g}(u_{j+1}) = \tilde{g}(\tilde{g}(u_j)) \neq u_j$ by the acyclic property of g .

It then follows by induction that $u_{j+2} \neq u_j$ for all $j \geq 0$.

We may construct a subsequence $u_{\sigma(0)} \prec u_{\sigma(1)} \prec \dots$, with $\sigma : \mathbb{N} \longrightarrow \mathbb{N}$ a monotonically increasing function, as follows. We let $\sigma(0) = 0$ and for each $j \geq 0$, and define

$$\sigma(j+1) = \begin{cases} \sigma(j) + 1, & \text{if } u_{\sigma(j)+2} = f(u_{\sigma(j)+1}) = f^2(u_{\sigma(j)}) \\ \sigma(j) + 2, & \text{otherwise} \end{cases} \quad (3.24)$$

for all $j \geq 0$. Suppose for some $j \geq 1$ that $u_{\sigma(j)+1} = f(u_{\sigma(j)})$.

- If $u_{\sigma(j)+2} = f(u_{\sigma(j)+1})$ as well, we have $u_{\sigma(j)} \prec u_{\sigma(j)+1} = u_{\sigma(j+1)}$, and we also have $u_{\sigma(j+1)+1} = f(u_{\sigma(j+1)})$.
- Otherwise, we have $u_{\sigma(j+1)} = u_{\sigma(j)+2} = \tilde{g}(u_{\sigma(j)+1})$, which implies that either $u_{\sigma(j)+2} = u_{\sigma(j)+1} = f(u_{\sigma(j)})$ or $u_{\sigma(j)+2} \sim u_{\sigma(j)+1} = f(u_{\sigma(j)})$. In either case, we again have $u_{\sigma(j)} \prec u_{\sigma(j+1)}$, and once more we have $u_{\sigma(j+1)+1} = f(u_{\sigma(j+1)})$.

As $u_{\sigma(0)+1} = u_1 = f(u_0)$, by induction we then have $u_{\sigma(k)} \prec u_{\sigma(k+1)}$ for all $k \geq j$. As $V(G)$ is finite, there must be integers N, N' such that $N < N'$ and $u_{\sigma(N)} = u_{\sigma(N')}$: then we have $u_{\sigma(N)} \preceq u_{\sigma(N+1)} \preceq \dots \preceq u_{\sigma(N')} = u_{\sigma(N)}$, where $u_{\sigma(N)} \neq u_{\sigma(N+1)}$ as noted above. Then \preceq is not antisymmetric.

If on the contrary (f, \preceq) is a modified flow, and (g, \leq) is also a modified flow for some partial order \leq , it follows that $f = g$. Thus (G, I, O) and M have at most one modified flow (f, \preceq) , for \preceq the natural pre-order of f . ■

The uniqueness of a modified flow (f, \preceq) , when we require that \preceq be the natural pre-order of f , will play an important role in describing a semantic map for the DKP and RBB constructions. Different “flow-functions” f correspond to different sets of dependencies in a 1WQC procedure: provided a procedure \mathfrak{P} in which some of the dependencies have been obfuscated or eliminated altogether by Pauli simplifications and signal shifting, it may be difficult to determine whether there exists a successor function which describes the actual dependencies which are present. This problem is simplified if we are promised that there is at most one successor function f yielding a modified flow (f, \preceq) : we may simply attempt to construct f , and determine whether the dependencies in \mathfrak{P} are consistent with those described by f .

3.3 Decompositions of geometries

The proof of Lemma 3.3 suggests an approach to determining when a tuple (G, I, O, M) has a modified flow, by means of a decomposition of the geometry (G, I, O) into sub-geometries. Examining such decompositions will lead to an efficient algorithm for obtaining the candidate successor function f associated with a modified flow, which we may use to implement a semantic map for the DKP and RBB constructions.

3.3.1 Star geometries and star decompositions

For a geometry (G, I, O) , note that the natural pre-order of an injective function $f : O^c \rightarrow I^c$ is characterized entirely by the transitive closure of local relations of vertices in G . Similar local relations are used in the proof of Lemma 3.3 to define star graphs which are used to define both star circuits, and their representations as procedures in the 1WQC model. For an injective function $f : O^c \rightarrow I^c$, we may characterize these local relations in terms of the following structures:

Definition 3.5. Let (G, I, O) be a geometry, and $f : O^c \rightarrow I^c$ an injective function. For $v \in V(G)$ define the vertex-set

$$V_{v\star} = \left\{ w \in V(G) \mid w \in \{v, f(v)\} \text{ or } [w \sim f(v) \text{ and } w \neq f(f(v))] \right\}. \quad (3.25)$$

We then define the *star geometry rooted at v* (or *centered at $f(v)$*) is the geometry $(G_{v\star}, I_{v\star}, O_{v\star})$ given by

$$G_{v\star} = G[V_{v\star}] - \{xw \mid x, w \neq f(v)\}, \quad (3.26a)$$

$$I_{v\star} = V_{v\star} \setminus \{f(v)\}, \quad (3.26b)$$

$$O_{v\star} = V_{v\star} \setminus \{v\}. \quad (3.26c)$$

Star geometries correspond to star circuits (as described in Section 3.1.1) and $\mathcal{Z}z$ operations, via their representations as 1WQC procedures. Consider the 1WQC procedure $\mathfrak{P} = \Phi(C_\star[u_1, \dots, u_n \frac{w}{v}])$ obtained from a stable-index expression for star circuit C_\star as in (3.1). Then \mathfrak{P} has a geometry (G, I, O) consisting of a star graph G with center $w \in O \setminus I$, and where $v \in I \setminus O$. If f is the index successor function of the stable-index expression, we may then show that (G, I, O) is the star geometry with root v and center w as in Definition 3.5. We may similarly show that $\Phi(\mathcal{Z}z)$ also gives rise to a star geometry, where the mediating qubit $a = f(a)$ is both the center and the root of the star geometry. (The same also holds for $\Phi(\mathcal{Z}z_d)$, for the mapping Φ and unitary $\mathcal{Z}z_d$ defined in the proof of Lemma 3.3.)

Star geometries $(G_{u\star}, I_{u\star}, O_{u\star})$ are thus precisely those geometries which underlie the 1WQC procedures $\Phi(C_\star)$, where C_\star are the featured circuits in decompositions described in Section 3.1.1. Extending the results of [12], we may consider how such structures for a geometry (G, I, O) may be used to describe “candidate” circuits C for 1WQC procedures \mathfrak{P} , where C may correspond to \mathfrak{P} via the DKP or RBB constructions: we consider how to construct such circuits in Section 3.4. Motivated by the connection with circuit constructions, we may consider a decomposition of geometries similar to the compositions of simple circuits in Section 3.1.1:

Lemma 3.5. *Let (G, I, O) be a geometry and $M \subseteq O^c$. Then (G, I, O, M) has a modified flow (f, \preceq) if and only if there is a decomposition of (G, I, O) into star geometries rooted at each of the vertices $v \in O^c$, together with the geometry $(G[\tilde{I}], I, \tilde{I})$, where $\tilde{I} = V(G) \setminus \text{img}(f)$.*

Proof. We proceed by induction on the size of $\text{img}(f)$. If $\text{img}(f) = \emptyset$, then $\text{dom}(f) = \emptyset$ as well, in which case $I \subseteq V(G) = O$; we then have $(G, I, O) = (G[\tilde{I}], I, \tilde{I})$. The natural pre-order in this case is the equality relation, which is antisymmetric; then $(f, =)$ is a modified flow. The Lemma then holds for $\text{img}(f) = \emptyset$.

For the induction step, suppose that $|\text{img}(f)| = N + 1$ for some $N \in \mathbb{N}$, and that the claim holds for geometries (G, I, O) with flows (f', \preceq) in which $|\text{img}(f')| \leq N$.

- Suppose that (G, I, O, M) has a modified flow: in particular, the natural pre-order \preceq is a partial order. Because $V(G)$ is finite, there is a vertex $v \in V(G)$ such that $v \prec w$ only for w maximal in the natural pre-order \preceq . Let $\mathcal{G}_{v\star} = (G_{v\star}, I_{v\star}, O_{v\star})$, and $\mathcal{G}' = (G', I, O')$, where

$$G' = G \setminus f(v), \quad \text{and} \quad (3.27a)$$

$$O' = \left\{ \begin{array}{ll} O, & \text{if } v = f(v) \\ O \triangle \{v, f(v)\}, & \text{otherwise} \end{array} \right\}. \quad (3.27b)$$

(Again, $A \triangle B$ denotes the symmetric difference of A and B .) Note that $O_{v\star} \subseteq O$ by definition, and either $I_{v\star} = O_{v\star} \triangle \{v, f(v)\} \subseteq O \triangle \{v, f(v)\}$ (if v and $f(v)$ differ) or $I_{v\star} = O_{v\star} \subseteq O$ (if $f(v) = v$). Because $V(G_{v\star}) = I_{v\star} \cup \{f(v)\}$ in both cases, we then have $V(G') \cap V(G_{v\star}) \subseteq I_{v\star} \cap O'$: then the composition

$(\bar{G}, \bar{I}, \bar{O}) = \mathcal{G}_{v\star} \circ \mathcal{G}'$ is well-defined, and we have

$$\begin{aligned}\bar{G} &= G_{v\star} \cup G' \\ &= \{xw \in G \mid w = f(v)\} \cup [G \setminus f(v)] = G;\end{aligned}\tag{3.28a}$$

$$\bar{I} = (I_{v\star} \setminus O') \cup I = \emptyset \cup I = I;\tag{3.28b}$$

$$\bar{O} = O_{v\star} \cup (O' \setminus I_{v\star}) = O_{v\star} \cup (O \setminus O_{v\star}) = O.\tag{3.28c}$$

Then, we have $(G, I, O) = \mathcal{G}_{v\star} \circ \mathcal{G}'$.

The restriction f' of the flow-function f to $V(G) \setminus O'$ does not include $f(v)$ in its image, and therefore has cardinality at most N ; by recursively applying the decomposition above, \mathcal{G}' has a decomposition into star geometries and the geometry $(G' \setminus \text{img}(f'), I, V(G') \setminus \text{img}(f'))$, where in particular $V(G) \setminus \text{img}(f') = V(G) \setminus \text{img}(f) = \bar{I}$, by the definition of f' . Thus, for any tuple (G, I, O, M) with a modified flow, (G, I, O) has a decomposition into star geometries as above.

- For the converse, suppose that $(G, I, O) = \mathcal{G}_{v\star} \circ \mathcal{G}'$ for some star geometry $\mathcal{G}_{v\star} = (G_{v\star}, I_{v\star}, O_{v\star})$ and a residual geometry $\mathcal{G}' = (G', I, O')$. If the latter has a star decomposition, and if f' is the restriction of f to $V(G')$, then (f', \preceq) is a modified flow for \mathcal{G}' by hypothesis as $\text{img}(f') = \text{img}(f) \setminus \{f(v)\}$. By assumption, $\mathcal{G}_{v\star}$ and \mathcal{G}' are composable, in which case we have $V(G_{v\star}) \cap V(G_{v\star}) = I_{v\star} \cap O'$. If $v \prec w$ for some $w \in V(G)$, we then have either $w = f(v)$, $w \in O' \setminus v$, or $w \succ w'$ for either $w' = f(v)$ or $w' \sim f(v)$. By construction, we have either $O = O' \setminus v$ in the case that $v = f(v)$, or $O = (O' \setminus \{v\}) \cup \{f(v)\}$ otherwise; then for any $w' = f(v)$ or $w' \sim f(v)$, we have $w' \in O$. As such w' lie outside of the domain of f , they are maximal in the natural pre-order; then $w \succ v$ only if $w \in O$. In particular, if the restriction of the natural pre-order to \mathcal{G}' is such that (f', \preceq) is a modified flow, then (f, \preceq) is also a modified flow. Thus, the existence of a star decomposition for (G, I, O) implies that it has a modified flow, by induction. ■

Thus, we may decompose a circuit with a modified flow into star geometries. Figure 10 illustrates a decomposition of a geometry with a flow into star geometries in the manner described above; Figure 11 illustrates a more general case for a modified flow. Also illustrated in these figures are “candidate circuit” constructions for unitary circuits, which we describe in Section 3.4.

3.3.2 Structural characterization of star decompositions

In order to define a semantic map \mathcal{S} for the DKP and RBB constructions, it will be necessary to attempt to obtain a star decomposition of a geometry which is initially not known to have a modified flow. We may facilitate this by showing that we may characterize star decompositions independently of modified flows, as follows:

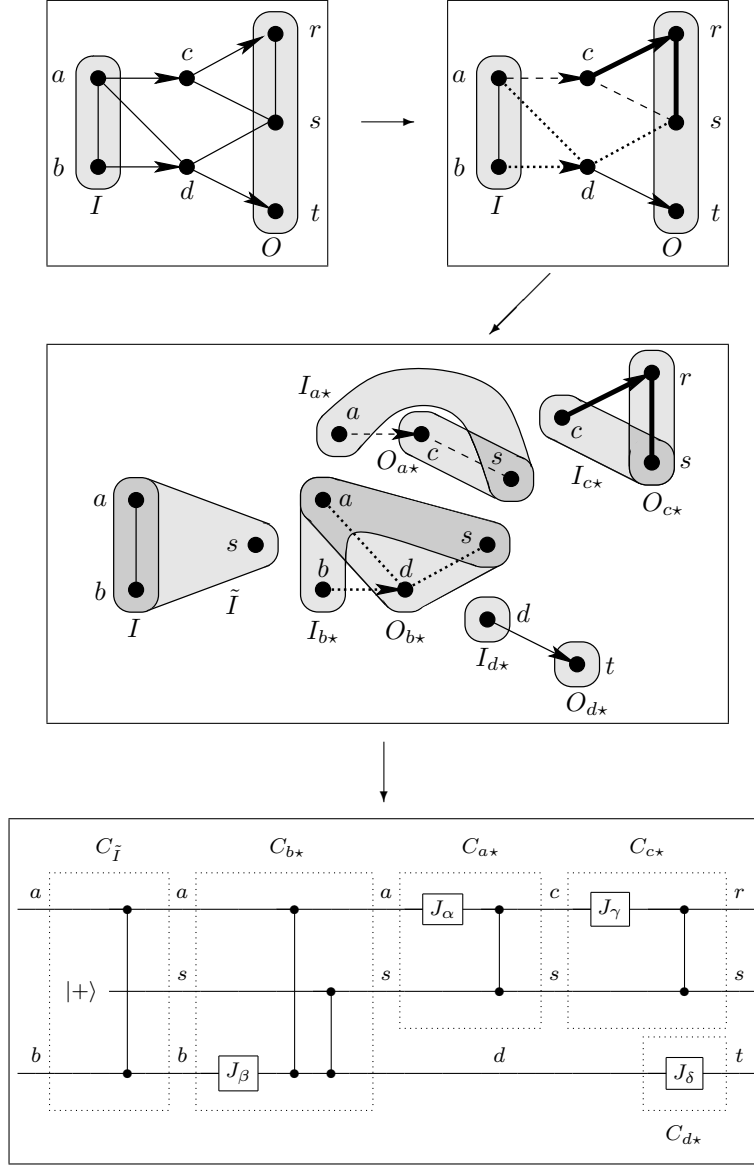


Figure 10: Illustration of a decomposition of a geometry (G, I, O) with a flow (f, \preceq) into star geometries for each $v \in O^c$, and a reduced geometry $(G[\tilde{I}], I, \tilde{I})$ for $\tilde{I} = V(G) \setminus \text{img}(f)$. In the upper three panels, the action of f is represented by arrows; different styles of edges indicate edges belonging to different star geometries. The middle panel illustrates each star geometry in the decomposition separately (with vertices repeated between different star geometries), adjacent to the geometries with which they may be composed. The bottom panel illustrates the unitary circuits corresponding to these star geometries (including a labelling of wire-segments corresponding to stable tensor indices), and the complete circuit obtained from composing them.

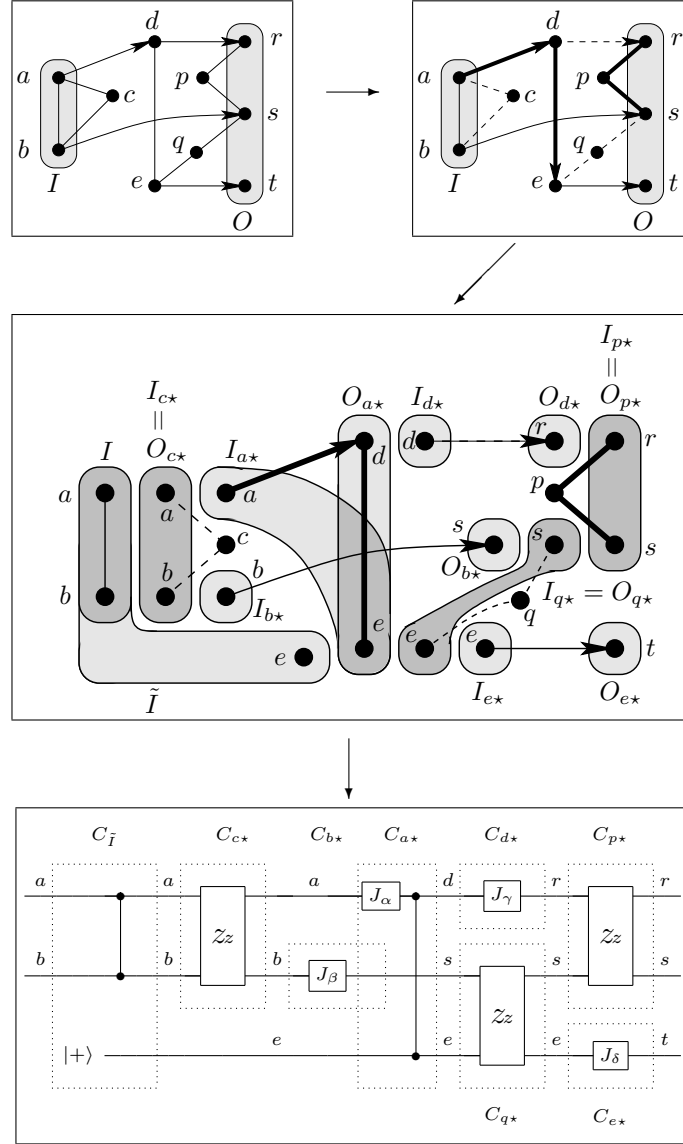


Figure 11: Illustration of a decomposition of a geometry (G, I, O) with a modified flow (f, \preceq) into star geometries for each $v \in O^c$, and a reduced geometry $(G[\tilde{I}], I, \tilde{I})$ for $\tilde{I} = V(G) \setminus \text{img}(f)$. In the upper three panels, the action of f is represented by arrows. Different styles of edges indicate edges belonging to different star geometries (as are edges of the same style, but which are disconnected from each other). Arrows are omitted for $f(v) = v$. The middle panel illustrates each star geometry in the decomposition separately (with vertices repeated between different star geometries), adjacent to the geometries with which they may be composed. The bottom panel illustrates the composition of the unitary circuits corresponding to the star geometries (including a labelling of wire-segments corresponding to stable tensor indices).

Lemma 3.6. *Let (G, I, O) be a geometry and $M \subseteq O^c$. Suppose that (G, I, O) can be expressed as $\mathcal{G}_{\ell\star} \circ \mathcal{G}_{(\ell-1)\star} \circ \cdots \circ \mathcal{G}_{1\star} \circ \mathcal{G}_{\tilde{I}}$, where $\mathcal{G}_{\tilde{I}} = (G[\tilde{I}], I, \tilde{I})$ for some subset $\tilde{I} \subseteq V(G)$ with $|\tilde{I}| = |O| = |V(G)| - \ell$, and where for each $0 \leq j < \ell$ the geometry $\mathcal{G}_{j\star} = (G_j, I_j, O_j)$ satisfies either*

- (i) $I_j = O_j = V(G_j) \setminus \{v_j\}$, for some $v_j \in V(G) \cap M$ which is adjacent to every other element of $V(G_j)$;
- (ii) $V(G) = I_j \cup O_j$, $I_j = (I_j \cap O_j) \cup \{v_j\}$, and $O_j = (I_j \cap O_j) \cup \{w_j\}$, for some distinct $v_j, w_j \in V(G)$ such that w_j is adjacent to every other element of $V(G_j)$.

Let $f : O^c \longrightarrow I^c$ be defined so that $f(v_j) = v_j$ for any v_j as in the first case above, and $f(v_j) = w_j$ for any v_j, w_j as in the second case; and let \preceq be the natural pre-order for f . Then (f, \preceq) is a modified flow for (G, I, O, M) .

Proof. Let f be defined as above, and let \preceq be its natural pre-order. By construction, each geometry \mathcal{G}_j is then a star geometry with respect to f in the sense of Definition 3.5, whether or not (f, \preceq) is a modified flow. We may show by induction on $\ell = |V(G)| \setminus |O|$ that \preceq is a partial order.

- If $\ell = 0$, then $O = V(G)$, in which case $\text{dom}(f) = \emptyset$. Then the natural pre-order is the equality relation, which is antisymmetric and therefore a partial order.
- Suppose the Lemma holds for all geometries with fewer vertices than (G, I, O) , but with an output subsystem of the same size. By hypothesis, we have $(G, I, O) = (\mathcal{G}_\ell, I_\ell, O_\ell) \circ \mathcal{G}'$, where we may write

$$\mathcal{G}' = (G', I, O') = \mathcal{G}_{(\ell-1)\star} \circ \cdots \circ \mathcal{G}_{1\star} \circ \mathcal{G}_{\tilde{I}}; \quad (3.29)$$

in particular, we have $V(G') \cap V(G_\ell) \subseteq I_\ell \cap O'$. If f' is the restriction of f to O' , and \preceq' is the restriction of \preceq to $V(G)$, the pair (f', \preceq') is then a modified flow for \mathcal{G}' .

- (i) Suppose $f(v_\ell) = v_\ell$: then $V(G) \setminus V(G') = \{v_\ell\}$, and we have $w \in O' = O$ for every $w \sim v_\ell$. Then the only relations that \preceq contains that \preceq' does not are $u \preceq v_\ell$ for qubits such that $f(u) \in O$, and $v_\ell \preceq w$ for $w \in O$; that is, $u \preceq v_\ell \preceq w$ for some w maximal and u bounded above in \preceq' only by maximal elements.
- (ii) Suppose $f(v_\ell) = w_\ell \neq v_\ell$: then $v_\ell \in O' \setminus O_\ell$, and $w_\ell \in O_\ell \setminus O'$. Then the only relations that \preceq contains that \preceq' does not are $u \preceq w_\ell$ for qubits such that $f(u) = w$ such that $w \sim w_\ell$, and $v_\ell \preceq w$ for qubits $w \sim w_\ell$. In either case, we have $w_\ell \in O'$; thus we only add additional (maximal) upper bounds for u bounded only bounded by maximal elements, and new upper bounds for v_ℓ which are also maximal.

In either case, \preceq is a partial order; thus (f, \preceq) is a modified flow. ■

We will call a decomposition as in Lemma 3.6 a *star decomposition* of a geometry (G, I, O) , and the associated geometries \mathcal{G}_j in the decomposition *star geometries*, whether or not they are defined with respect to a corresponding modified flow (f, \preceq) . More generally, we will call a geometry (G_*, I_*, O_*) a *star geometry* when the input and output subsystems satisfies either property (i) or property (ii) in the statement of Lemma 3.6 above. In the case that $I_* = O_*$, we say that unique vertex $v \in V(G_*) \setminus O_*$ is both the root and the center of (G_*, I_*, O_*) ; otherwise, we say that the unique vertex $v \in I_* \setminus O_*$ is the root of (G_*, I_*, O_*) , and the unique vertex $w \in O_* \setminus I_*$ is the center.

The proof of Lemma 3.6 suggests a recursive decomposition of a geometry (G, I, O) into star geometries using “maximal” star geometries, in the following sense:

Definition 3.6. For a geometry (G, I, O) , a *maximal star geometry* is a star geometry (G_*, I_*, O_*) such that $(G, I, O) = (G_*, I_*, O_*) \circ (G', I, O')$ for some sub-geometry (G', I, O') . We call the latter the *residual geometry* of this decomposition.

We may characterize a maximal star geometry as follows:

Lemma 3.7. Let (G_*, I_*, O_*) be a maximal star geometry, with root v and center w , for some geometry (G, I, O) . Then one of the following two properties hold:

- (i) $v = w \in I^c \cap O^c$, and is adjacent in G only to elements of O ;
- (ii) $v \neq w$, $w \in O$, and $v \in O^c$ is the only neighbor of w in G which is not in O .

Conversely, for a geometry (G, I, O) and vertices $v, w \in V(G)$ for which either of the above hold, the geometry (G_{v*}, I_{v*}, O_{v*}) such that $G_{v*} = G[W]$ for W consisting of w and its neighbors in G , with input and output subsystems $I_{v*} = V(G_{v*}) \setminus \{w\}$ and $O_{v*} = V(G_{v*}) \setminus \{v\}$, is a maximal star geometry for (G, I, O) .

Proof. We first show that any maximal star geometry has either property (i) or property (ii). By the definition of a maximal star geometry, we have $O_* \subseteq O$; as $v \notin O_*$, we have $v \in O^c$. If $v = w$, by definition v is adjacent to every element of $I_* = O_*$, and every neighbor of v in G is an element of I_* ; thus v is adjacent only to elements of O . Also, in the case that $v = w$, we have $v \notin I_*$ by definition, in which case $v \in I^c$ as well. Otherwise, we have $w \in O_* \subseteq O$ and $v \in I_* \setminus O_*$, so that $v \in O^c$. Also, if $v \neq w$ by definition, all edges incident to w in G are of the form uw , where either $u = v$ or $u \in O_* \subseteq O$; then v is the unique element of O^c to which w is adjacent in G .

Conversely, suppose that v and w are vertices satisfying either property (i) or property (ii), and let (G_{v*}, I_{v*}, O_{v*}) be the geometry described. Let $G' = G \setminus w$ and $O' = O \setminus \{w\}$. In either case, we may show that $(G, I, O) = (G_{v*}, I_{v*}, O_{v*}) \circ (G', I, O')$, and that (G_{v*}, I_{v*}, O_{v*}) is also a star geometry. The Lemma then holds. ■

By definition, any geometry with a star decomposition has a maximal star geometry; we may then attempt to obtain a star decomposition by recursively constructing maximal star geometries for successive residual geometries. In the next section, we show how this leads to an efficient algorithm for obtaining a star decomposition (and a modified flow) for an arbitrary geometry (G, I, O) and subset $M \subseteq O^c$, when one exists.

3.3.3 Efficiently decomposing geometries

In order to be able to test the consistency of measurement dependencies of a 1WQC procedure \mathfrak{P} with those arising in the DKP or (simplified) RBB construction, we may attempt to determine whether the tuple (G, I, O, M) associated to \mathfrak{P} has a modified flow. As a special case, we may efficiently determine when a geometry (G, I, O) has a flow, and construct one if one exists. The most efficient known algorithm to do this is that of Mhalla and Perdrix [14]. We show how their algorithm may be adapted to produce modified flows when they are present, using an analysis in terms of maximal star geometries.

Maximally delayed flows. We may summarize the algorithm of [14] as follows. To find a flow (f, \preceq) or a geometry (G, I, O) , we consider the length of the longest chains of each qubit from the output set, and construct “layer sets” consisting of vertices of similar depth:

Definition 3.7 ([14], Definitions 4 & 5). For a geometry (G, I, O) and a flow (f, \preceq) , let $V_0^{\preceq}(G)$ be the set of vertices of $V(G)$ which are maximal in \preceq , and let $V_{j+1}^{\preceq}(G)$ be the maximal set of vertices of the set

$$W_j^{\preceq} = V(G) \setminus \left(\bigcup_{\ell=0}^j V_\ell^{\preceq} \right) \quad (3.30)$$

in the order \preceq , for any $j > 0$. A flow (f, \preceq) is *more delayed* than another flow (f', \preceq') if we have $|W_j^{\preceq}| \leq |W_j^{\preceq'}|$ for all $j \in \mathbb{N}$, where this inequality is strict for at least one such j ; and (f, \preceq) is *maximally delayed* if no flow for (G, I, O) is more delayed.

A maximally delayed flow is also a flow of minimum depth, *i.e.* that the order \preceq of such a flow has the same depth as the natural pre-order for the given flow-function f . For a maximally delayed flow (f, \preceq) , let $\ell : V(G) \rightarrow \mathbb{N}$ be the function mapping each vertex $v \in V(G)$ to the integer j such that $v \in V_j^{\preceq}$. Then, the partial order \preceq characterized by $v \preceq w \iff [v = w \text{ or } \ell(v) > \ell(w)]$ also yields a minimum-depth flow by construction.

Mhalla and Perdrix show for a maximally delayed flow (f, \preceq) for (G, I, O) that $V_0^{\preceq} = O$, and V_1^{\preceq} is the set of elements of W_0^{\preceq} such that, for some $w \in O$, v is the only neighbor of w which is contained in W_0^{\preceq} . By induction, augmenting the set O to include successive “layers” V_j^{\preceq} , one may show that

$$V_{j+1}^{\preceq} = \left\{ v \in W_j^{\preceq} \mid \exists w \in V(G) \setminus W_j^{\preceq} : N_G(w) \cap W_j^{\preceq} = \{v\} \right\}, \quad (3.31)$$

where $N_G(w)$ is the set of vertices adjacent to w in G . By a depth-first search from O , we may find in a finite number of iterations of j those qubits $w \in V(G) \setminus W_j^{\preceq}$ which have such a neighbor v , and accumulate them into new layers V_{j+1}^{\preceq} .

Relation to maximal star geometries. This decomposition of the geometry (G, I, O) into layers V_j^{\preceq} corresponds strongly to the star decomposition of a geometry (G, I, O) .

The maximal elements of $V(G)$ in the natural pre-order are exactly the elements of O (as $z \preceq f(z)$ for any $z \in O^c$): thus, the root v of a maximal star geometry in (G, I, O) is one such that all neighbors $z \sim f(v)$ are elements of O or equal to v itself. The set V_1^{\prec} as described above is the set of precisely those vertices which are the root of a maximal geometry. Thus, the algorithm of [14] may be described as finding flows by performing a star decomposition of (G, I, O) when it has a flow.

We may easily extend the maximal delay principle of [14] for flows to one which identifies a maximal star geometry for (G, I, O) in the more general case where this geometry (together with a set M) has a modified flow. For such a triple (G, I, O, M) with a modified flow (f, \preceq) , we may define the sets V_j^{\prec} and W_j^{\prec} as in Definition 3.7. Then for each $j \in \mathbb{N}$, V_{j+1}^{\prec} consists of those vertices $v \in W_j^{\prec}$ such that, for any $w \in V(G)$ such that $w = f(v)$ or $w \sim f(v)$, we have $w = v$ or $w \in V(G) \setminus W_j^{\prec}$. For $v \notin M$, this implies that (as in the analysis of [14]) the only neighbor of $f(v)$ which lies in W_j^{\prec} is v itself; for $v \in M$, it may also be possible that v has no neighbors in W_j^{\prec} . The flow-finding algorithm of [14] can then be easily extended to test for the latter condition, yielding a procedure to produce a star decomposition of (G, I, O) when one exists, as we describe below.

Bounding the search for roots of maximal star geometries. The flow-finding algorithm of Mhalla and Perdrix ([14], Algorithm 1) operates on the basis of maintaining a set of “correcting vertices” w which, for each new layer V_{j+1}^{\prec} , may potentially be the successor of some vertex $v \in V_{j+1}^{\prec}$. These correcting vertices are precisely the centers of maximal geometries, in which case the corresponding v is a root. We thus take a similar approach to obtain a star decomposition by “removing” layers of maximal star geometries, identified with their centers.

For modified flows, we must accommodate vertices $v \in M$ which may be simultaneously the root and the center of a geometry. To ensure that we do not fruitlessly examine vertices of M at each layer, we maintain a digraph whose arcs point in either direction along each edge of G , and from which we delete any arc ending at a vertex which has already been indicated as the root of a star geometry. Then, if a potential center w has a single neighbor v which has not been matched to a star geometry, v is the root and w the center of a maximal star geometry; and if a vertex w has not yet been identified as the center of a star geometry only neighbors vertices which have already been identified as centers, w is both the center and the root of a maximal star geometry. As a result, we may easily determine whether a potential center w has a single neighbor v which has not been matched to a star geometry (in which case v is the root and w the center of a maximal star geometry), or whether a vertex which has not yet been identified as the center of a star geometry only neighbors vertices which have already been identified as centers (in which case w is both the center and the root of a maximal star geometry).

Algorithm 2 defines a subroutine `RemoveStarGeom` to perform the necessary arc deletions for the digraph described above whenever we define the layer of a vertex, as well as several other operations useful to an iterative procedure for discovering the maximal star geometries of a star decomposition.

Data : M , a subset of specially designated vertices
 W , the set of vertices not yet marked as roots of star geometries
emptyLayer, a flag indicating if the present layer is empty
 D , a digraph in which vertices $v \in V(G) \setminus W$ have in-degree 0
 $f : V(G) \longrightarrow V(G)$ a mapping from “roots” to “centers”
 S' , a set of possible centers of star geometries for the next layer
 $v \in V(G)$, a vertex to mark as the root of a star geometry
 $w \in V(G)$, a vertex to mark as the center associated to v

Effect: Assigns $f(v) \leftarrow w$, adds v to the set of potential centers for the next layer and to the ordered list of qubits with identified star geometries, removes all of the inbound-arcs to v in the digraph D , and checks the neighbors of v in D for vertices in M with out-degree 0 to mark as a center for the next layer.

```

1 subroutine RemoveStarGeom( $v, w$ ) :
2   Set emptyLayer  $\leftarrow$  false.
3   Remove  $v$  from the set  $W$ , and set  $f(v) \leftarrow w$ .
4   Insert  $v$  to the beginning of  $L$ .
5   For every neighbor  $z$  with an outbound arc to  $v$  in  $D$ , do:
6     Remove the outbound arc  $z \rightarrow v$  from  $D$ .
7     If  $z \in M \cap W$  and  $z$  has no more outbound arcs, then add  $z$  into  $S'$ .
8   endfor

```

ALGORITHM 2. A subroutine to attribute a vertex to the current layer, for use in an iterative procedure for finding the roots of maximal sub-geometries with a modified flow. In particular, this subroutine maintains the arcs of a digraph D (a “sub-digraph” of the graph G of a geometry, interpreted as a symmetric digraph) from which we remove any arc inbound to a vertex with a well-defined layer.

Accumulating roots of star geometries. Using RemoveStarGeom as a subroutine, we may easily define a procedure similar to Algorithm 1 of [14] which either (a) determines that a star decomposition of the geometry does not exist, implying the non-existence of a modified flow, or (b) constructs a modified flow, represented as a tuple (f, L) consisting of a partial function $f : V(G) \longrightarrow V(G)$ mapping the root of each star geometry to the corresponding center, and a list L of the roots corresponding to a composition order for the star geometries.^v Algorithm 3 presents such a procedure, which we describe below.

We begin by creating the digraph D which governs the search for roots: we copy all of the neighborhood relations of G , omitting those arcs ending in O , as these cannot be roots of star geometries. Similarly, as no $w \in O$ is a root, it must lie outside the

^vThe list L constructed by Algorithm 3 is a linear order, *i.e.* a partial order of *maximum* depth. To find one of *minimum* depth as in Algorithm 1 of [14], it suffices to maintain a layer counter and to record the layer for which each vertex $v \in O^c$ is the root of a maximal star geometry. However, as we are ultimately interested in producing a (linearly ordered) circuit expression, it is not necessary to obtain a partial order of smaller depth.

Input : G , a graph with subsets I, O, M , where $M \subseteq V(G) \setminus O$ consists entirely of vertices with default measurement angles $\pi/2$.
Output: Either `nil`, if (G, I, O, M) does not have a modified flow; or an ordered pair (f, L) consisting of a partial function $f : V(G) \rightarrow V(G)$ mapping roots of star geometries to their respective centers, and a sequence L of the roots representing a composition order for star geometries.

```

1  procedure FindModStarDecomp( $G, I, O, M$ ) :
2      Let  $f : V(G) \rightarrow V(G)$  be a mapping from “roots” to “centers”.
3      Let  $L$  be a sequence of vertices (initially empty).
4      Let  $W \leftarrow \emptyset$  be a set of potential roots of star geometries.
5      Let  $S \leftarrow \emptyset$  be a set of potential centers of maximal star geometries.
6      Let  $D$  be the completely disconnected digraph on  $V(G)$ .
7      For all  $v \in V(G)$ , do:
8          For every  $w \in N_G(v) \setminus O$ : add the arc  $v \rightarrow w$  to  $D$ .
9          If  $v \in O$ , then
10             Set  $f(v) \leftarrow \text{nil}$ .
11             If  $v \notin I$ , then add  $v$  into  $S$ .
12         else
13             Add  $v$  into  $W$ .
14         endif
15     endfor
16     Let emptyLayer be a flag indicating failure to find any new roots.
17     Repeat:
18         Set emptyLayer  $\leftarrow$  true, initially.
19         Let  $S' \leftarrow \emptyset$  be the set of potential centers for the next iteration.
20         For each  $w \in S$ , do:
21             If  $w \in W$ , then
22                 Call RemoveStarGeom( $w, w$ ).
23             else, if  $D$  has only one outbound arc  $w \rightarrow v$ , then
24                 Call RemoveStarGeom( $v, w$ ).
25                 If  $v \notin I$ , then add  $v$  into  $S'$ .
26             else
27                 Add  $w$  back into  $S'$ .
28             endif
29         endfor
30         Set  $S \leftarrow S'$  for the next iteration.
31     until emptyLayer.
32     If  $W = \emptyset$ , then return  $(f, L)$ ; else return nil.

```

ALGORITHM 3. An iterative algorithm to discover a star decomposition inspired by the analysis of [14], using the subroutine `RemoveStarGeom` of Algorithm 2.

domain of f , so we set $f(w) \leftarrow \text{nil}$; but if $w \in O$ is not an element of I , it will be the center of some star geometry, so we insert it to a set S of candidates for centers of *maximal* star geometries. Any vertex $w \notin O$ is the root of some star geometry in a star decomposition of (G, I, O) , provided such a decomposition exists; we then insert such w into a set W of candidates for roots of maximal star geometries.

We then search for layers of new roots, corresponding to the layer-sets in the analysis of [14]. For each layer, we iterate through the set of potential centers for maximal star geometries, which initially includes no elements of W . We maintain the invariant that D contains all (directed) adjacency relations of G , except that it contains no arcs ending at vertices $w \notin W$. We also maintain the invariant that $v \in W$ becomes an element of S only if it is an element of M and has out-degree zero in D , in which case it is both the root and the center of a maximal star geometry. We indicate the removal of vertices (as a result of decomposing the geometry into star geometries and an as-yet decomposed “residue”) by absence from both the sets S and W ; thus S represents at each stage the output subsystem of the residual geometry.

In any iteration, for the center w of any maximal star geometry, either w has a unique neighbor $v \notin W$ in G (in which case $w \rightarrow v$ is the only outbound arc from w in D), or $w \in M$ and has no neighbors in W (in which case it has out-degree 0 in D).

- In the latter case, unless w is isolated in G , there was some final neighbor u which was removed from W in an earlier iteration; immediately after the removal of u from W , we may then identify that w is the root (and center) of a maximal star geometry. In the call to `RemoveStarGeom(u, u')`, we thus examine each of the neighbors $w \in M \cap W$ (restricting our search to qubits in M which have not yet been assigned as the root to a previously removed star geometry) which have an outbound arc $w \rightarrow u$, to see if that arc is the last outbound arc from w . If this is the case for some w , we add w into the set of potential centers S' for the next iteration.

In Algorithm 3, this is the only condition under which an element of W may be inserted into S' ; the only other condition under which S' accumulates elements is for a vertex v which has already been removed from W . Thus, in the subsequent iteration, any element of $S \cap W$ is necessarily the root (and center) of a maximal star geometry, which we may then decompose from the parent geometry. This decomposition is represented by the removal of w from W , and the omission of w from S' .

- In the case where w has a unique outbound arc $w \rightarrow v$ in D , the vertex v is necessarily the root of a maximal star geometry for which w is the center. We may then decompose this star geometry from the parent geometry, which we represent by the removal of v from W (and the insertion of v into S'), and the omission of w from S' . In the next iteration, as an element of the output subsystem of the residual geometry, v is then a potential center of a maximal star geometry.

For any $w \in S$ which is not the center of a maximal geometry as above, we carry w forward for the next layer by including w in S' . We assign the list of new candidate successors to S before the end of the iteration; and we repeat until we cannot find any more roots for some layer.

If we cannot assign any vertices to new layers, this is either because we have exhausted the possible roots (in which case $W = \emptyset$), or because there are no vertices which are reachable from the existing layers which have viable successors. In the latter case, there is no modified flow; and in the former, we return the tuple (f, L) .

Run-time analysis of Algorithm 3. The run-time of Algorithm 3 is similar to that of Algorithm 1 of [14], which we may show as follows. In the following, we let $n = |V(G)|$, $k = |O|$, and $m = |E(G)|$. We assume that the graph G and digraph D are represented with adjacency lists for each vertex, in the latter case using lists of neighbors from inbound and outbound arcs separately. The partial function f is represented with an array structure taking the value `nil` (indicating a vertex not in the domain) or vertices in $V(G)$. We may represent the sets M and W as arrays of pointers to nodes of a linked-list structure, enabling constant-time membership checking and element insertion, as well as linear-time traversal of the elements of the set. The sets S and S' we represent simply as linked lists, which also provides us with constant-time initialization to the empty set.

In the subroutine `RemoveStarGeom`(v, w), the run-time is dominated by the loop over the neighbors z of v , which requires time at most $O(\deg_G(v))$. As this subroutine is called at most once for each vertex v , the total run-time of `RemoveStarGeom`(v, w) over the execution of `FindModStarDecomp` is $O(\sum_v \deg_G(v)) = O(m)$.

Initially, S has k elements, none of which are elements of W ; and in each iteration, as every element $w \in S \setminus W$ gives rise to the removal of some vertex v from W and the potential insertion of v into S' , there always remain at most k elements of $S \setminus W$ at each iteration of the loop on lines 17–31. The comparisons on lines 21 and 23 can both be performed in constant time, as can the set-inclusions on lines 25 and 27. The assignment on line 30 can be performed simply by re-attributing the element list of S' to S . There are at most n iterations of the loop on lines 17–31, as it only repeats as long as at least one vertex has been assigned to a new layer. Then, setting aside the work performed for vertices $w \in S \cap W$ in the loop on lines 17–31 and the work performed by `RemoveStarGeom`, the work performed by that loop is $O(kn)$.

As elements of W are only included into S' when they are guaranteed to be attributed to the next layer, the total work performed by that loop (again aside from work performed by `RemoveStarGeom`) for elements of M is then $O(n)$. The run-time of `FindFlow`(G, I, O, M) is then $O(kn + m)$, dominated by the work performed by `RemoveStarGeom`(v, w) for each $w \in V(G)$, and by the iteration of the loop on lines 17–31 for at most n layers.

3.3.4 Extremal analysis of geometries with modified flows

The run-time of $O(kn + m)$ for Algorithm 3 is precisely the complexity of Algorithm 1 of [14]. However, Mhalla and Perdrix also use an extremal result presented in [15] which bounds the number of edges m in a geometry with a flow by $kn - \binom{k+1}{2}$. This allows any geometry with edges in excess of this to be rejected at the outset, so that the run-time for geometries satisfying this bound may be simplified to $O(kn)$. By observing the correspondence drawn in Section 3.1.1 between successor functions and

the ordering described in (2.14) for indices of stable-index expressions, we may extend this extremal result as follows:

Lemma 3.8. *Let (G, I, O) be a geometry such that (G, I, O, M) has a modified flow, for some vertex set $M \subseteq V(G)$. If $|V(G)| = n$, $|E(G)| = m$, and $|O| = k$, then $m \leq nk - \binom{k+1}{2}$. Furthermore, there exists a geometry with a modified flow which saturates this bound.*

Proof. Consider a decomposition $(G, I, O) = (G_{v\star}, I_{v\star}, O_{v\star}) \circ (G', I, O')$ into a maximal geometry and a residual geometry. If v is both the root and the center of the geometry, then it is adjacent to at most k vertices, as it is adjacent only to elements of $O' = O$. Otherwise, let $w = f(v)$ be the center of $(G_{v\star}, I_{v\star}, O_{v\star})$; then w is again adjacent only at most k vertices, as it is adjacent to v , and otherwise only to elements of $O \setminus \{w\}$. In either case, $(G_{v\star}, I_{v\star}, O_{v\star})$ contributes at most k edges to (G, I, O) . In either case, we have $|O'| = |O| = k$.

By induction, each star geometry in a decomposition of (G, I, O) contributes at most k edges, and has an output subsystem of size k . Then the output subsystem $\tilde{I} = V(G) \setminus \text{img}(f)$ of the initial geometry $(G[\tilde{I}], I, \tilde{I})$ in the star decomposition also contains k vertices; this geometry then has at most $\binom{k}{2} = \frac{1}{2}k(k-1)$ edges. Furthermore, as $(G[\tilde{I}], I, \tilde{I})$ does not contain the center of any star geometry, and as each star geometry has a center distinct from the others, there are at most $n - k$ star geometries in the decomposition, each with a distinct center. The number of edges in (G, I, O) is then bounded by

$$(n - k)k + \frac{1}{2}k(k - 1) = nk - \frac{1}{2}k^2 - \frac{1}{2}k = nk - \binom{k+1}{2}, \quad (3.32)$$

as required; this bound is saturated by the extremal construction of [15]. \blacksquare

As a result, we may simplify the run-time of Algorithm 3 by $O(kn)$ in this case as well, again by a pre-processing stage which returns `nil` for geometries with more than $kn - \binom{k+1}{n}$ edges.

3.4 Candidate circuit constructions

Following the approach in the proof of Lemma 3.3, we may consider how to construct “candidate” circuits C for 1WQC procedures \mathfrak{P} , by decomposing of the underlying geometry into star geometries and mapping each star geometry to a corresponding unitary circuit. This will give rise to a circuit C over the gate set \mathcal{B}_{DKP} or \mathcal{B}_{RBB} , such that $\Phi(C)$ is another 1WQC procedure with the same underlying geometry. If we can prove that the circuit C performs the same operation as \mathfrak{P} , then C describes the semantics of \mathfrak{P} in the unitary circuit model. We may use this as the final step towards defining a semantic map \mathcal{S} for the DKP and (simplified) RBB constructions.

3.4.1 Star circuits corresponding to star geometries

For a star geometry, we may consider a corresponding star circuit, possibly parameterized (in the case of a star geometry with root v and center $w = f(v) \neq v$) by an angle

related to the measurement angle θ_v of v :

$$C_{v\star} = \left(\prod_{\substack{u \sim w \\ u \neq v}} \wedge Z[u, w] \right) J(-\theta_v) \left[\frac{w}{v} \right] . \quad (3.33)$$

For star geometries whose center and root are the same (in the case that $f(v) = v$), corresponding to the mediating qubits of a $\mathcal{Z}\mathcal{Z}_d$ operation for some $d \geq 2$ as in (3.13), the corresponding star circuit is just the operator $\mathcal{Z}\mathcal{Z}_d[u_1, \dots, u_d]$ acting on the input/output subsystems: this corresponds to the decomposition of $\mathcal{Z}\mathcal{Z}_d$ as

$$\mathcal{Z}\mathcal{Z}_d[w_1, \dots, w_d] \propto P \left[\frac{a}{a} \right] \left(\prod_{j=1}^d \wedge Z[a, w_j] \right) |+\rangle \left[\frac{a}{a} \right] , \quad (3.34)$$

generalizing (3.2). We may also correspond the geometry $(G[\tilde{I}], I, \tilde{I})$ to a circuit consisting solely of $\wedge Z$ gates, as described with (3.20):

$$C_{\tilde{I}} = \left(\prod_{vw \in E(G[\tilde{I}])} \wedge Z[v, w] \right) \left(\prod_{v \in \tilde{I} \setminus I} |+\rangle \left[\frac{v}{v} \right] \right) ; \quad (3.35)$$

as described in Section 3.1.1, such a circuit arises in star decompositions of circuits as well. We may then construct a candidate circuit C as in the preceding section by re-composing the circuits corresponding to the geometries in the star decomposition of (G, I, O) . Such re-composed circuits are illustrated in the bottom panels of Figure 10 and Figure 11. For instance, in Figure 10, we may recover the geometry in the upper left-hand panel by composing

$$(G, I, O) = (G_{c\star}, I_{c\star}, O_{c\star}) \circ (G_{d\star}, I_{d\star}, O_{d\star}) \circ (G_{a\star}, I_{a\star}, O_{a\star}) \circ (G_{b\star}, I_{b\star}, O_{b\star}) \circ (G[\tilde{I}], I, \tilde{I}) ; \quad (3.36)$$

and similarly, the larger circuit C in Figure 10 is obtained by composing

$$C = C_{c\star} \circ C_{d\star} \circ C_{a\star} \circ C_{b\star} \circ C_{\tilde{I}} , \quad (3.37)$$

where the circuits $C_{u\star}$ are those corresponding to each star geometry, and $C_{\tilde{I}}$ corresponds to the geometry $(G[\tilde{I}], I, \tilde{I})$.

3.4.2 The gate model for candidate circuit constructions

Neither of the circuits described in Figure 10 or Figure 11 are generated over the gate set $\{H, T, \wedge Z\}$; and the latter is not defined over either \mathcal{B}_{DKP} or \mathcal{B}_{RBB} , but instead over the more general gate-set $\tilde{\mathcal{B}}$ described in the proof of Lemma 3.3. (As we suggested just after the proof of that Lemma 3.3, this gate-set corresponds naturally to the structure of tuples (G, I, O, M) with modified flows; we may refine this statement and describe this set of gates as that which naturally corresponds to star geometries. In order to use

star decompositions to describe a semantic map corresponding to representations of circuits over $\{H, T, \wedge Z\}$, we must first translate the circuits from the gate-set \mathcal{B} .

There is one apparent problem: the circuits arising from such a decomposition are more general than those in the domain of Φ . This may conceivably complicate the task of comparing the circuits arising from a semantic map \mathcal{S} based on star decompositions, and the circuits which lie in the domain of either \mathcal{R}_{DKP} or \mathcal{R}_{RBB} . We may show that this more general sort of construction will not arise in practise for 1WQC procedures resulting from either the DKP construction or the RBB construction when $|I| = |O|$. In this case, a geometry has at most one star decomposition up to commuting terms, by the uniqueness of the modified flow (f, \preceq) shown in Theorem 3.4 for this case. In particular:

- for (G, I, O) which underly procedures \mathfrak{P} produced by the DKP construction, there will be no qubits $v \in O^c$ such that $f(v) = v$, because (f, \preceq) will be a flow;
- for (G, I, O) which underly procedures \mathfrak{P} produced by the RBB construction, the only edges vw in (G, I, O) for which $v \neq f(w)$ and $w \neq f(v)$ are those such that either $v = f(v)$ or $w = f(w)$: *i.e.* where one of the qubits is a mediator qubit for a $\mathcal{Z}\mathcal{Z}$ operation between two logical qubits.

Thus, while the candidate circuits for arbitrary geometries with modified flows may be more general, the candidate circuits for 1WQC procedures \mathfrak{P} arising from a circuit C over \mathcal{B}_{DKP} or \mathcal{B}_{RBB} will be restricted to the same gate-set as C itself when $|I| = |O|$.

For a semantic map defined relative to a particular construction, we may also ensure that we obtain candidate circuits over the appropriate gate-set by imposing restrictions on the type of star geometries which we permit in the decomposition of a particular geometry. However, as we considering only the special case $|I| = |O|$, this will not be necessary for our analysis.

3.4.3 Relationships between 1WQC procedures and candidate circuits

Properly speaking, a candidate circuit is a construction from a 1WQC procedure \mathfrak{P} rather than from a star decomposition of a geometry (G, I, O) , as the angular parameters of the $J(\theta)$ gates in a candidate circuit are determined by the measurement angles in \mathfrak{P} . In order to use candidate circuits as a means of defining a semantic map \mathcal{S} , we must consider the relationship between 1WQC procedures and their candidate circuits.

By Theorem 3.1, we may characterize the dependencies of operations in procedures \mathfrak{P} obtained from either the DKP or RBB constructions, relative to a candidate for the extended successor function for a corresponding circuit. If the geometry (G, I, O) underlying \mathfrak{P} (together with the set M of mediating qubits) has a modified flow (f, \preceq) , this defines a candidate circuit C , in which the gates $J(\alpha_u)$ arising from star-circuits $C_{u\star}$ (for a qubit $u \neq f(u)$ at the root of a star geometry) have their angles fixed by $\alpha_u = -\theta_u$, where θ_u is the default measurement angle of u in \mathfrak{P} .

By the observations made above, the circuit C will be defined over the gate set \mathcal{B}_{DKP} or \mathcal{B}_{RBB} , depending on whether \mathfrak{P} arises from the DKP or RBB construction: we may then consider the 1WQC procedure $\bar{\mathfrak{P}} = \bar{\Phi}(C)$, which will coincide with $\Phi(C)$. The procedure $\bar{\mathfrak{P}}$ will have the same geometry as \mathfrak{P} , and the same default measurement

angles: if it also has the same correction and measurement dependencies, this implies that \mathfrak{P} performs the same transformation as the circuit C .

For procedures \mathfrak{P} arising from the DKP or RBB constructions in the case that $|I| = |O|$, there is only one modified flow for (G, I, O) , which must then be the extended successor function for the circuit over the \mathcal{B}_{DKP} or \mathcal{B}_{RBB} gate sets produced in the course of constructing \mathfrak{P} . It follows then that C is also precisely the circuit arising in the construction of \mathfrak{P} , in which case the normal form of $\Phi(C)$ will be congruent to \mathfrak{P} . Thus, to obtain a semantic map for the DKP construction when $|I| = |O|$, it suffices to translate C into the gate-set $\{H, T, \wedge Z\}$. For the RBB construction, to obtain a circuit with complexity no greater than the original circuit from which \mathfrak{P} was produced, we must also simplify C by removing products of the form $J(0)J(0)$ and $J(\pi/2)J(\pi/2)J(\pi/2)$ introduced by the construction of Section A.2.2.

In the case that $|I| \neq |O|$, even conditioned on \mathfrak{P} arising from the DKP or RBB constructions, it may be difficult to obtain the modified flow function f corresponding to the circuit arising in the construction of \mathfrak{P} , and therefore to produce a suitable candidate circuit. Without an algorithm that can construct such candidate circuits in the case that $|I| = |O|$, for instance by finding the modified flow (f, \preceq) for which \mathfrak{P} satisfies Theorem 3.1, we must therefore restrict ourselves to the DKP and RBB constructions in the case where no fresh qubits are introduced, *i.e.* which perform unitary bijections.

3.4.4 Efficient construction of candidate circuits over $\{H, T, \wedge Z\}$

For the sake of completeness, we present a procedure `BuildCircuit` (extending the remarks made in Section III B of [12]) for constructing a stable-index representation for a candidate circuit for \mathfrak{P} , represented in the gate set $\{H, T, \wedge Z\}$ (rather than the sets \mathcal{B}_{DKP} or \mathcal{B}_{RBB}). The procedure `BuildCircuit` takes as input a representation of a star decomposition of a geometry (G, I, O) : we may represent this in terms of the graph G of the geometry, the successor function f from the modified flow which defines the star decomposition, and a sequence L of the roots of the component star geometries. This procedure is presented in Algorithm 4.

For each root u in the list L , `BuildCircuit` determines whether it is the root of a geometry whose root and center are both u (corresponding to a $\mathcal{Z}_{\text{ZZ}_d}$ gate acting on the neighbors of u in G), or where the center $v = f(u)$ differs from u (corresponding to a star circuit as in (3.33)). It then appends to a circuit expression C the appropriate sequence of operations:

- By expanding $Z = \mathbb{1} - 2|1\rangle\langle 1|$ in the tensor product $Z \otimes \cdots \otimes Z$, and considering the expansion of $\mathcal{Z}_{\text{ZZ}_d} = e^{-i\pi Z^{\otimes d}/4}$ in terms of multi-qubit operations $e^{i(|1\rangle\langle 1|^{\otimes j})\theta}$ for $0 \leq j \leq d$, we may show that

$$\mathcal{Z}_{\text{ZZ}_d}[w_1, \dots, w_d] \propto \left[\prod_{j=1}^n T^2[w_j] \right] \left[\prod_{\substack{j,k=1 \\ j < k}}^n \wedge Z[w_j, w_k] \right]. \quad (3.38)$$

We may construct such a gate for w_j ranging over all neighbors of a qubit $u = f(u)$ by obtaining the set N of these neighbors, and for each $w \in N$, adding

Input : G , the a graph for the geometry \mathcal{G} of a 1WQC procedure
 $f : V(G) \longrightarrow V(G)$, a modified flow function for \mathcal{G}
 L , a list of roots of star geometries in a decomposition of \mathcal{G}
 $t : V(G) \longrightarrow \mathbb{Z}$ an array denoting multiples of $\pi/4$
Output: A candidate circuit generated over $\{H, T, \wedge Z\}$, corresponding to the star decomposition described by G , f , and L , with powers of T gates given by the parameters stored in $t(u)$.

```

1  procedure BuildCircuit( $G, f, L, t$ ) :
2      Let  $C$  a stable-index representation of a circuit, initially empty.
3      For each  $u \in L$ , do:
4          If  $u = f(u)$ , then
5              Let  $N \leftarrow N_G(u)$ .
6              For each  $w \in N$ , do:
7                  For each  $w' \in N$ : add  $\wedge Z[w, w']$  to  $C$ .
8                  Add  $T[w] T[w]$  to  $C$ .
9                  Remove  $w$  from  $N$ .
10             endfor
11         else
12             Let  $v \leftarrow f(u)$ .
13             Let  $N \leftarrow N_G(u) \setminus \{u, f(v)\}$ .
14             For each  $w \in N$ : add  $\wedge Z[v, w]$  to  $C$ .
15             Add  $H[\frac{v}{u}]$  to  $C$ .
16             If  $t(u) < 0$ 
17                 then for  $j = 1$  to  $-t(u)$ : add  $T^\dagger[u]$  to  $C$  ;
18                 else for  $j = 1$  to  $t(u)$ : add  $T[u]$  to  $C$  .
19             endif
20         endfor
21     Let  $\tilde{I} \leftarrow V(G) \setminus \text{img}(f)$ .
22     For each  $u \in \tilde{I}$ , do:
23         For each  $w \in N_G(u)$ , do:
24             If  $w \in \tilde{I}$ , then add  $\wedge Z[u, w]$  to  $C$ .
25         endfor
26         If  $u \notin I$ , then add  $|+\rangle[\frac{u}{u}]$  to  $C$ .
27         Remove  $u$  from  $\tilde{I}$ .
28     endfor
29     return  $C$ .

```

ALGORITHM 4. A procedure to construct a candidate circuit corresponding to a star decomposition of a geometry, realized over the gate-set $\{H, T, \wedge Z\}$.

T operations on w to C , as well as controlled- Z operations between w and the other elements of N . Removing w from N after doing so will then prevent controlled- Z operations from being added twice for each pair of neighbors.

- For each qubit u such that $u \neq f(u)$, we may obtain the set of N neighbors of $v = f(u)$ belonging to the output subsystem $O_{u\star}$ of the star geometry $(G_{u\star}, I_{u\star}, O_{u\star})$. N will then consist of the neighbors of v excluding u and (if it is defined) $f(v) = f(f(u))$. We may then add the star circuit $C_{u\star}$ to C by adding $\wedge Z$ operations between each $w \in N$ and v , followed by the components of the gate $J(\theta)$ for the appropriate angle θ . For angles θ constrained to integer multiples of $\pi/4$, this will consist of $H[\frac{v}{u}] T^{t(u)}[u]$ for some integer array t given as input: if $t(u)$ is negative, we may add products of $T^\dagger[u]$ to C .

After having added the star circuits for each star geometry, we then add the circuit corresponding to the geometry $(G[\tilde{I}], I, \tilde{I})$ for $\tilde{I} = V(G) \setminus \text{img}(f)$. After constructing the set \tilde{I} , we iterate through the qubits $u \in \tilde{I}$, adding to C the $\wedge Z$ operations acting on u and (if necessary) representing the fact that $u \notin I$ by adding a preparation of a fresh qubit $|+\rangle$ in order to advance u as a fresh index in the circuit C .^w Subsequently removing u from \tilde{I} prevents duplicate $\wedge Z$ operations from being added to C .

Run-time analysis of Algorithm 4. We may bound the run-time of `BuildCircuit` as follows. Let $n = |V(G)|$, $m = |E(G)|$, and $k = |O|$. We assume a list-like structure for the stable-index expression C and for the sets $N_G(u)$, and an array-structure for f , where $f(v) = \text{nil}$ denotes that $v \notin \text{dom}(f)$.

- For each qubit $u = f(u)$, the number of gates to add corresponding to the expression for $\mathcal{Z}_{\deg_G(u)}$ above is $O(\deg_G(u)^2)$. As the output subsystem of each star geometry in a star decomposition has the same size input and output subsystem, we have $\deg_G(u) \leq |O| = k$ for each such u ; then the total work number of gates arising from these qubits is $O(k^2n)$.
- For each qubit $u \neq f(u)$, the number of gates to add corresponding to the circuit $C_{u\star}$ is $O(\deg_G(f(u)))$; summing over all such u , we have a total of $O(m)$ gates arising from these qubits.
- For each qubit $u \in \tilde{I} = V(G) \setminus \text{img}(f)$, we have $\deg_G(u)$ gates to add, plus possibly an initial state $|+\rangle[\frac{u}{u}]$. Summed over all $u \in \tilde{I}$, we again have at most $O(m)$ operations arising in this case.

As each operation requires time $O(1)$ to add, the run-time of Algorithm 4 is then $O(k^2n + m)$.

^wWhile the semantic map we will consider does not involve the introduction of fresh qubits, such fresh qubits do not complicate the process of constructing candidate circuits, and so we include such preparations in Algorithm 4.

3.5 Complete algorithm for the semantic map when $|I| = |O|$

We may now describe a single, complete procedure `Semantic` (presented in Algorithm 5) to perform a semantic map \mathcal{S} with respect to the representations \mathcal{R}_{DKP} and \mathcal{R}_{RBB} represented by the constructions of Section 2.3. We may describe the operations

<p>Input : \mathfrak{P}, a 1WQC procedure in normal form.</p> <p>Output: Either <code>nil</code>, if \mathfrak{P} is not (congruent to) the normal form of a procedure as in Lemma 3.3, or a stable-index representation of a circuit C generated over the gate-set $\{H, T, \wedge Z, Z_{zz_d}\}_{d \geq 2}$ which performs the same transformation as \mathfrak{P}.</p> <pre> 1 procedure Semantic(\mathfrak{P}) : 2 Let (G, I, O) be the geometry underlying \mathfrak{P}. 3 Let $n \leftarrow V(G)$, $m \leftarrow E(G)$, $k \leftarrow O$. 4 Let M be a set of possible mediator qubits, initially empty. 5 Let $t : O^c \rightarrow \mathbb{R}$ be an array of multiples of $\pi/4$. 6 For each operation $M_u^{\theta;\beta}$ in \mathfrak{P}, do: 7 Set $t(u) \leftarrow -4\theta/\pi$. 8 If $\theta = \pi/2$, then add u into M. 9 endfor 10 If $m > nk - \binom{k+1}{2}$, then return <code>nil</code>. 11 Let <code>starDecomp</code> \leftarrow FindModStarDecomp(G, I, O, M). 12 If <code>starDecomp</code> = <code>nil</code> 13 then return <code>nil</code>; 14 else let $(f, L) \leftarrow$ <code>starDecomp</code>. 15 If TestDependencies(\mathfrak{P}, f) = <code>false</code>, then return <code>nil</code>. 16 Let $C \leftarrow$ BuildCircuit(G, f, L, t). 17 return RemoveIdops(C). </pre>
--

ALGORITHM 5. A complete algorithm for a semantic map with respect to either the DKP or simplified RBB constructions, using the procedures defined in Algorithm 1, Algorithm 4, and Algorithm 3, as well as a procedure `RemoveIdops` which performs trivial simplifications of products of single-qubit gates.

performed by `Semantic` in three different phases, which we may describe as follows.

Obtain the underlying structure of \mathfrak{P} . We first obtain the geometry (G, I, O) underlying \mathfrak{P} , as well as the set M of qubits measured with default angles $\pi/2$, in order to obtain a star decomposition. We also obtain the cardinalities $n = |V(G)|$, $m = |E(G)|$, and $k = |O|$. For each qubit u measured, we record the integers $t(u) \leftarrow -4\theta/\pi$ which corresponds to the power of T for which $HT^{t(u)} = J(-\theta_u)$; if the measurement at u corresponds to the execution of a $J(\theta)$ gate in the candidate circuit, we may use $t(u)$ to determine the expansion of this gate in the gate-set $\{H, T, \wedge Z\}$.

Rather than perform these operations literally as presented in Algorithm 5, we may obtain this data with a single pass of the operations of \mathfrak{P} , constructing G , I^c , and O^c by inserting qubits and edges as each qubit is acted on by preparation maps, entangling operations, and measurements; we may obtain n , m , and k similarly. Using arrays of pointers to linked list nodes to represent I^c and O^c , we may then construct a similar representation of both I and O with a single traversal of $V(G)$. As we do so, we also assign values to the array t and insert qubits into the set M . As \mathfrak{P} has at most $O(n)$ preparation, measurement, and correction operations, and $O(m)$ entangling operations, it has size $O(m + n)$: this is then the time required for this phase of the procedure.

Examine the geometry and the measurement dependencies of \mathfrak{P} . As we noted in Section 3.3.4, a 1WQC procedure cannot have a modified flow if it has more than $nk - \binom{k+1}{2}$ edges, in which case it lies outside the range of both the DKP and RBB constructions by Lemma 3.3; we then return `nil`. We then attempt to obtain a star decomposition for (G, I, O) with the set M of potential mediating qubits: if this fails, we also return `nil`.

By Theorem 3.1, we may characterize the dependencies of operations in procedures obtained from either the DKP or RBB constructions, relative to a candidate for the successor function for a corresponding circuit. For the star decomposition described by the ordered pair (f, L) returned in the previous step, there is a corresponding candidate circuit C for which f is an extended index successor function, as described in Section 3.4. In the case that $|I| = |O|$, there is no other candidate successor function for (G, I, O, M) , by Theorem 3.4. Then the candidate circuit C described by (f, L) is the only one which describes a unitary transformation which may be performed by \mathfrak{P} .

The 1WQC procedure $\Phi(C)$ has underlying geometry (G, I, O) by construction; if \mathfrak{P} has the same operations (with the same dependencies) as the normalization of $\Phi(C)$, they then perform the same transformation, and thus \mathfrak{P} is a representation of C . Again by construction, the normalization of $\Phi(C)$ satisfies the conditions of Theorem 3.1 with respect to f . Then if `TestDependencies`(\mathfrak{P}, f) returns `true`, we may provide the semantics for \mathfrak{P} in terms of unitary circuits by constructing C ; otherwise, we have no such guarantee, and so we return `nil`.

We may compare m to $nk - \binom{k+1}{2}$ in time $\text{polylog}(n)$; and the respective runtimes of `FindModStarDecomp` and `TestDependencies` are $O(kn) \subseteq O(n^2)$ and $O(kn^2) \subseteq O(n^3)$, by the analyses of Section 3.3.3 and Section 3.1.2 respectively.

Constructing a circuit for \mathfrak{P} . Provided that the procedure does not return `nil` in the previous phase, the procedure \mathfrak{P} performs a unitary transformation, and in particular is a representation of the “candidate” circuit C obtained from the star decomposition of (G, I, O) . We therefore construct the circuit C , using the star decomposition described by (f, L) . We produce a stable-index representation for C , representing it in the gate-set $\{H, T, \wedge Z\}$ rather than the set \tilde{B} used in Lemma 3.3: this takes time $O(k^2n + m) = O(k^2n)$, as $m \leq kn - \binom{k+1}{2}$ by hypothesis.

In the case of procedures \mathfrak{P} obtained by applying the RBB construction to some original circuit \tilde{C} , we must perform some final simplifications. We call a final subroutine `RemoveIdops`(C) in order to remove superfluous powers of T which may

arise from the conversion of operations $\mathcal{Z}z = (T^2 \otimes T^2)^\wedge Z$, and which also removes products of the form $H^2 = J(0)J(0)$ and $HT^2HT^2HT^2 = J(\pi/2)J(\pi/2)J(\pi/2)$ on individual qubits such as those introduced in the stable-index construction presented in Section A.2.2. As the dependencies of \mathfrak{P} satisfies the constraints of Theorem 3.1 relative to the successor function f , this circuit provides the semantics for \mathfrak{P} as a unitary circuit, by the analysis of page 61.

The list L produced by `FindModStarDecomp` contains roots of star-geometries in a non-increasing order (in terms of the natural pre-order \preceq for the successor function f it produces), and the operations of C are in a similar order from the construction of `BuildCircuit`(G, f, L, t). We may therefore perform this simplification of C in a single pass from right to left, using buffers for each logical qubit of the circuit to maintain accumulated sequences of single-qubit operations performed between pairs of two-qubit operations. We may then cancel products of single-qubit unitaries that occur at the end of the buffer, as appropriate. The amount of time required to do so is then a constant factor times the size of the circuit, which by the analysis of page 62 is also $O(k^2n + m) = O(k^2n)$.

Remarks on circuit congruence and total run-time. In order to consider the correctness of Algorithm 5, we may consider without loss of generality circuits \tilde{C} represented in the gate-sets \mathcal{B}_{DKP} or \mathcal{B}_{RBB} which do not contain redundant operations $J(0)J(0)$ or $J(\pi/2)J(\pi/2)J(\pi/2)$, and which do not introduce fresh qubits. We may proceed as follows.

If we apply the DKP construction to such a circuit \tilde{C} over the gate-set \mathcal{B}_{DKP} , we obtain a procedure \mathfrak{P} for which the qubits of O^c are labelled by deprecated indices in \tilde{C} , and I^c by advanced indices; where the entanglement graph is the same as the interaction graph of \tilde{C} ; and where the default measurement angles are the additive inverses of the angles of the $J(\theta)$ gates. Furthermore, the geometry (G, I, O) has a modified flow (f, \preceq) , in which f is the index successor function of \tilde{C} . Correspondingly, the candidate circuit C for \mathfrak{P} has an interaction graph given by the entanglement graph of \mathfrak{P} , and a successor function which agrees with f . The single-qubit gates corresponding to the edges $uf(u)$ in the interaction graph correspond to operations $J(-\theta_u)$, where θ_u is the default measurement angle of u ; and all other edges correspond to $^\wedge Z$ operations. We may then define a bijective mapping between the terms of \tilde{C} and the terms of C , via the structures induced in and recreated from \mathfrak{P} .

For the RBB construction, the analysis is similar: the only changes are the fact that the edges of the interaction graph of \tilde{C} which correspond to operations $\mathcal{Z}z$ must be subdivided by a single vertex to yield the entanglement graph of \mathfrak{P} , and paths of the form $J(0)J(0)$ or $J(\pi/2)J(\pi/2)J(\pi/2)$ may be inserted to achieve the topological constraint of being an induced subgraph of a two-dimensional grid. In constructing the candidate circuit C , the subdivided edges corresponding to $\mathcal{Z}z$ operations are remerged, and the superfluous products of single-qubit unitaries are once more removed. In this case, we may then also define a bijective mapping between the terms of \tilde{C} and the terms of C , via the structures induced in and recreated from \mathfrak{P} .

Thus, `Semantic` defines a semantic map for both the DKP and (simplified) RBB

constructions, restricted to unitary bijections. The run-time of `Semantic` is dominated by the time required to compare the dependencies of \mathfrak{P} against the criteria of Theorem 3.1, which is $O(nm) \subseteq O(kn^2)$.

4 Extensions and limitations

The analysis of 1WQC procedures presented in Section 3, in terms of modified flows and star decompositions, may be extended in a natural way to more general formulations of the 1WQC model. At the same time, this approach has limitations in its ability to describe even some of the earliest known constructions for 1WQC computation. In this section, we will describe in brief some of these extensions and limitations.

4.1 Extending beyond measurement in the XY plane

The model of measurement-based computation which we have focused on in this article (following the treatments in [1] and [2]) is one in which all single-qubit measurements are performed in the XY plane, *i.e.* with respect to uniform superpositions of the eigenstates of the Pauli Z operator. A natural extension of this is to allow measurements in any basis in a “Pauli plane”, *i.e.* for any uniform superposition of the eigenstates of X and Y as well. We may define the notation $|\pm\mathbf{xy}_\theta\rangle = |\pm_\theta\rangle$ (denoting the fact that these states lie in the XY plane), and then define

$$|\pm\mathbf{yz}_\theta\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \pm \frac{e^{-i\theta}}{2}(|0\rangle - |1\rangle), \quad (4.1a)$$

$$|\pm\mathbf{xz}_\theta\rangle = \frac{1}{2}(|0\rangle + i|1\rangle) \pm \frac{e^{i\theta}}{2}(|0\rangle - i|1\rangle), \quad (4.1b)$$

which denote bases lying in the YZ and XZ planes respectively. Then, we define the measurement operators

$$\begin{aligned} M_a^{(XY,\theta)}(\rho) &= \langle +\mathbf{xy}_\theta | \rho | +\mathbf{xy}_\theta \rangle_a \otimes |0\rangle\langle 0|_{s[a]} \\ &\quad + \langle -\mathbf{xy}_\theta | \rho | -\mathbf{xy}_\theta \rangle_a \otimes |1\rangle\langle 1|_{s[a]}, \end{aligned} \quad (4.2a)$$

$$\begin{aligned} M_a^{(YZ,\theta)}(\rho) &= \langle +\mathbf{yz}_\theta | \rho | +\mathbf{yz}_\theta \rangle_a \otimes |0\rangle\langle 0|_{s[a]} \\ &\quad + \langle -\mathbf{yz}_\theta | \rho | -\mathbf{yz}_\theta \rangle_a \otimes |1\rangle\langle 1|_{s[a]}, \end{aligned} \quad (4.2b)$$

$$\begin{aligned} M_a^{(XZ,\theta)}(\rho) &= \langle +\mathbf{xz}_\theta | \rho | +\mathbf{xz}_\theta \rangle_a \otimes |0\rangle\langle 0|_{s[a]} \\ &\quad + \langle -\mathbf{xz}_\theta | \rho | -\mathbf{xz}_\theta \rangle_a \otimes |1\rangle\langle 1|_{s[a]}. \end{aligned} \quad (4.2c)$$

We may then consider extensions of the analysis of Section 3, in measurement-based models which include operations of the form above.

4.1.1 Stabilizers, certificates of unitarity, and local graph structure

Recall from Lemma 3.3 that we may construct procedures performing unitary transformations in the 1WQC model by performing the following operation after each measurement $M_u^{(XY,\theta)}$:

- If $\theta \neq \pi/2$, we perform an X operation on some specially designated qubit $v \neq u$ (which we denote $f(u)$ for each u), and a Z operation on each of the neighbors of v except for u itself, if the state of u is projected onto $|-xy_\theta\rangle$.
- If $\theta = \pi/2$, either we perform the operation described above, or we perform a Z operation on each neighbor of u , if the state of u is projected onto $|-xy_\theta\rangle$.

In either case, the operation which is performed after each measurement may be described by an operator

$$B_u = X_{f(u)} \left(\prod_{w \sim f(u)} Z_w \right), \quad (4.3)$$

up to an operation acting on u itself (which we describe as being destroyed by the measurement and which thus cannot be carried out). This unitary operation is the Pauli group on $V(G)$, and in particular stabilizes the state $|\Psi_{G,I}\rangle$ defined by

$$|\Psi_{G,I}\rangle \langle \Psi_{G,I}| = E_G N_{I^c} (|\psi\rangle \langle \psi|) \quad (4.4)$$

for any $|\psi\rangle \in \mathcal{H}_2^{\otimes I}$. In particular, as the preparation maps N_v adjoin qubits prepared in the $+1$ eigenstate of the X operator, and the E_{vw} maps perform two-qubit $\wedge Z$ operations (which lie in the Clifford group) on pairs of qubits, the state of the system just prior to any measurements is a stabilizer code [5] $\mathcal{S}_{G,I}$ generated by the operators

$$K_v = X_v \prod_{w \sim v} Z_w = E_G(X_v). \quad (4.5)$$

The modified flow conditions may then be considered as a certificate that there exists an order in which the qubits of O^c may be measured so that one may use the stabilizer formalism [5] to simulate the post-selection of the state $|+xy_\theta\rangle$ for each measurement (by applying suitable corrections should the measurement yield $|-xy_\theta\rangle$ instead).

We may observe that the analysis of measurements by a single-qubit observable \mathcal{O}_v in the stabilizer formalism requires only that \mathcal{O}_v either commutes or anticommutes with every generator of the subgroup of the Pauli group which stabilizes the state of the system. We may observe that XY -plane measurements such as those above can be described by measurements with respect to observables

$$\mathcal{O}^{XY,\theta} = |+xy_\theta\rangle \langle +xy_\theta| - |-xy_\theta\rangle \langle -xy_\theta| = \cos(\theta)X + \sin(\theta)Y; \quad (4.6)$$

such operators always anticommute with the Pauli Z operator, and also anticommute with X when $\theta = \pi/2$. Then, a modified flow function $f : O^c \rightarrow I^c$ for a geometry (G, I, O, M) describes a mapping from u (with an associated observable $\mathcal{O}_u^{XY,\theta_u}$) to some qubit $v = f(u)$ such that K_v anticommutes with $\mathcal{O}_u^{XY,\theta_u}$. The constraints on the partial order \preceq ensure that, for any linear measurement order extending \preceq , the generators which stabilize the state just prior to the measurement of u consist of $K_{f(u)}$, and operators which all commute with u . Then, the evolution of the state-space system

under these measurements may be efficiently described via the stabilizer formalism as a transformation of stabilizer codes (although how individual states within these codes transform may be difficult to simulate). As the modified flow conditions ensure that each measurement observable has a corresponding generator with which it anticommutes, the overall transformation will then be unitary.

A straightforward extension of modified flows. The fact that qubits $u \in M$ may have either $u \sim f(u)$ or $u = f(u)$ stems from the fact that $\mathcal{O}^{XY, \theta_u} = Y$ in this case: that is, the measurement basis for such qubits lies in the intersection of the XY and YZ planes. The condition $u \sim f(u)$ in this case is common to all other measurement angles, as this condition suffices for treatment via the stabilizer formalism for observables which anticommute with Z . We may then extend the notion of modified flows as follows to also include other measurement observables which, like Y , anticommute with X :

Definition 4.1. For a tuple $(G, I, O, M_{XY}, M_{YZ})$ consisting of a geometry (G, I, O) and sets $M_{XY}, M_{YZ} \subseteq O^c$ such that $O^c = M_{XY} \cup M_{YZ}$, an *extended flow*^x is an ordered pair (f, \preceq) consisting of a function $f : O^c \longrightarrow I^c$, and a partial order \preceq on $V(G)$, such that the conditions

$$f(v) \sim v, \quad \text{for } v \in M_{XY} \setminus M_{YZ}; \quad (4.7a)$$

$$f(v) = v, \quad \text{for } v \in M_{YZ} \setminus M_{XY}; \quad (4.7b)$$

$$f(v) \sim v \text{ or } f(v) = v, \quad \text{for } v \in M_{YZ} \cap M_{XY}; \quad (4.7c)$$

$$v \preceq f(v); \quad \text{and} \quad (4.7d)$$

$$w \sim f(v) \implies v \preceq w \quad (4.7e)$$

hold for all $v \in O^c$ and $w \in V(G)$.

In the above definition, the sets M_{XY} and M_{YZ} correspond to qubits whose default measurement basis lies in the XY-plane or the YZ-plane, respectively; the intersection are those qubits whose measurement basis is the eigenbasis of the Y operator, which is common to both. Modified flows then correspond to the special case where $M_{YZ} \subseteq M_{XY}$.

As with modified flows, an extended flow (f, \preceq) for a tuple $(G, I, O, M_{XY}, M_{YZ})$ certifies that any procedure on the geometry (G, I, O) which performs the appropriate types of measurement will perform a unitary transformation, provided also that it performs the appropriate corrections (or measurement adaptations) in response to the measurements. To this effect, we may prove an extension of Lemma 3.3, as follows:

Lemma 4.1. Suppose (f, \preceq) is a modified flow for $(G, I, O, M_{XY}, M_{YZ})$. Let $v \sim w$ denote the adjacency relation of G , and let $\lambda : V(G) \longrightarrow \{XY, YZ\}$ indicate the plane of measurement for each qubit, with $\lambda(u) = XY$ or $\lambda(u) = YZ$ for $u \in M_{XY} \cap M_{YZ}$,

^xThe definition of extended flows here is a refinement of the definition presented in [9]: the definition there can be recovered by requiring M_{XY} and M_{YZ} to be disjoint, in which case the set T in Definition 3-17 of [9] corresponds to M_{YZ} .

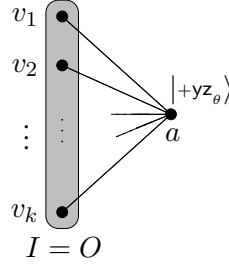


Figure 12: Illustration of the geometry for the procedure $\mathfrak{Z}\mathfrak{Z}\mathfrak{Z}_d^\theta$ defined in (4.10), together with the “default state” $|+\mathbf{y}\mathbf{z}_\theta\rangle$ to be measured for the qubit a in that procedure. (In the case that the measurement of a yields $|-\mathbf{y}\mathbf{z}_\theta\rangle$ instead, each element of $I = O$ is to be acted on with a Z operation.)

and $\lambda(u) = P$ for $u \in M_P$ otherwise. Then for any linear order \leq extending \preccurlyeq , the measurement procedure

$$\left[\prod_{u \in O^c} \leq \left(\prod_{\substack{w \sim f(u) \\ w \neq u}} Z_w^{s[u]} \right) \left(\prod_{\substack{v = f(u) \\ v \neq u}} X_v^{s[u]} \right) M_u^{(\lambda(u), \theta_u)} \right] E_G N_{I^c} \quad (4.8)$$

performs a unitary transformation.

Again, as we described above for modified flows, this may be proven for extended flows by simulating the evolution of the state-space via the stabilizer formalism, steering the state after each measurement to the one which would arise upon selection of the $|+\mathbf{x}\mathbf{y}_\theta\rangle$ or $|+\mathbf{y}\mathbf{z}_\theta\rangle$ result as appropriate.

Generalization of $\mathcal{Z}\mathcal{Z}_d$ operations using extended flows. An alternative proof would be to extend the analysis of the proof of Lemma 3.3 by considering the operation

$$\mathcal{Z}\mathcal{Z}_d^\theta = \exp(-i\theta Z^{\otimes d}/2) = e^{-i\theta \underbrace{[Z \otimes \cdots \otimes Z]}_{d \text{ times}}/2}, \quad (4.9)$$

generalizing the operation $\mathcal{Z}\mathcal{Z}_d$ defined in (4.9). We may define a 1WQC procedure

$$\mathfrak{Z}\mathfrak{Z}\mathfrak{Z}_{v_1, \dots, v_d}^{d, \theta} = \left(\prod_{j=1}^d Z_{v_j}^{s[a]} \right) M_a^{(\mathbf{y}\mathbf{z}, \theta)} \left(\prod_{j=1}^d E_{av_j} \right) N_a : \quad (4.10)$$

the geometry for this procedure is illustrated in Figure 12. Following the analysis of Section 3.8 of [19], we then have

$$\mathfrak{Z}\mathfrak{Z}\mathfrak{Z}_{v_1, \dots, v_d}^{d, \theta}(\rho_S) = [\mathcal{Z}\mathcal{Z}_d^\theta]_{v_1, \dots, v_d} \rho_S [\mathcal{Z}\mathcal{Z}_d^\theta]_{v_1, \dots, v_d}^\dagger. \quad (4.11)$$

The procedure of (4.10) is a straightforward generalization of the procedure $\mathfrak{Z}\mathfrak{Z}\mathfrak{Z}^d = \mathfrak{Z}\mathfrak{Z}\mathfrak{Z}^{d, \pi/2}$. In the same way that we map qubits u such that $u = f(u)$ to a circuit $C_u = \mathfrak{Z}\mathfrak{Z}\mathfrak{Z}_d$ acting on the neighbors of u for a modified flow as a part of a circuit construction, we may map qubits such that $u = f(u)$ to a circuit $\tilde{C}_u = \mathfrak{Z}\mathfrak{Z}\mathfrak{Z}_d^{\theta_u}$, where $|\pm yz_{\theta_u}\rangle$ is the default basis for the measurement of u .

Candidate circuits and star decompositions for extended flows. Just as geometries with modified flows correspond to circuits over the gate-set $\{J(\theta), \wedge Z, \mathfrak{Z}\mathfrak{Z}\mathfrak{Z}_d\}_{\theta \in \frac{\pi}{4}\mathbb{Z}, d \geq 2}$ via Lemma 3.3, we may define a map from circuits over $\{J(\theta), \wedge Z, \mathfrak{Z}\mathfrak{Z}\mathfrak{Z}_d^{\theta}\}_{\theta \in \frac{\pi}{4}\mathbb{Z}, d \geq 2}$ to the 1WQC model. For an extended flow, we may apply the same definition for star geometries as for modified flows: the only extension required in the correspondence between star decompositions and extended flows is that qubits $u \in M_{YZ} \setminus M_{XY}$ are necessarily both the root and the center of a star geometry with respect to an extended flow. It is then possible to show that star decompositions with respect to extended flows may be obtained with only minor modifications to the procedure `FindModStarDecomp` defined in Algorithm 3, and its subroutine `RemoveStarGeom` defined in Algorithm 2. In the former case, the set M on line 7 plays the role of $M_{YZ} \subseteq M_{XY}$ of qubits which may be both the root and center of a star geometry: it suffices them to replace M with M_{YZ} . In `FindModStarDecomp`, we must refine the condition on line 23, as there may not be qubits $w \in M_{YZ} \setminus M_{XY}$ which cannot be the center of a star geometry without also being the root: we must then revise the condition to require that

$$D \text{ has only one outbound arc } w \rightarrow v \text{ and } v \in M_{XY}, \quad (4.12)$$

the second condition of which was guaranteed for the special case of modified flows. Applying the same analysis as for Algorithm 3, this modified algorithm then finds a star decomposition for (G, I, O) , with respect to the constraints described by M_{XY} and M_{YZ} on the possible relationship between the roots and centers of each star geometry. In particular, as the proof of the extremal result of Lemma 3.8 carries forward for extended flows, the run-time of the modified algorithm is also $O(kn)$.

4.1.2 Generalized certificates of unitarity based on stabilizers

In the preceding section we observed one sense in which modified flows may be generalized, by observing that they essentially provide sufficient information to certify via the stabilizer formalism that a 1WQC procedure performs a unitary transformation. This is done by attributing some generator K_v to each qubit $u \in O^c$ to be measured, where the measurement on u may be described by some observable \mathcal{O}_u which anticommutes with K_v , but commutes with K_w for all $w \prec u$.

Similar, but more general, criteria that extended flows were defined in [4] in *generalized flows* and *Pauli flows*, which extend further to arbitrary generators of the initial stabilizer group $\langle K_w \rangle_{w \in I^c}$ (and thus to non-trivial products of the operators K_w) when XY, YZ, and XZ-plane measurements are all allowed. We may paraphrase the definition of Pauli flows as follows:

Definition 4.2. Let (G, I, O) be a geometry, and let $M_{XY}, M_{YZ}, M_{XZ} \subseteq O^c$ such that $O^c = M_{XY} \cup M_{YZ} \cup M_{XZ}$. Let \mathcal{P} denote the power-set function, and $\text{odd}_G(S)$ the

set of vertices in G which are adjacent to an odd number of elements of a set $S \subseteq V(G)$. Then a *Pauli flow* for $(G, I, O, M_{XY}, M_{YZ}, M_{XZ})$ is a tuple (g, \preceq) consisting of a function $g : O^c \rightarrow \mathcal{P}(I^c)$ and a partial order \preceq which satisfy the following conditions:^y

$$w \in g(v) \cup \text{odd}(g(v)) \implies v \preceq w, \quad (4.13a)$$

$$v \in \text{odd}(g(v)) \setminus g(v) \implies v \in M_{XY}, \quad (4.13b)$$

$$v \in g(v) \setminus \text{odd}(g(v)) \implies v \in M_{YZ}, \quad (4.13c)$$

$$v \in g(v) \cap \text{odd}(g(v)) \implies v \in M_{XZ}. \quad (4.13d)$$

From this definition, we recover extended flows if we require that the sets $g(u)$ be singleton sets for all u . (The definition of generalized flows may be recovered by requiring that M_{XY} , M_{YZ} , and M_{XZ} be disjoint.)

Based on these conditions, we may obtain a further extension of Lemma 3.3, which we paraphrase from Theorem 4 of [4] as follows:

Lemma 4.2. *Suppose (g, \preceq) is a Pauli flow for $(G, I, O, M_{XY}, M_{YZ}, M_{XZ})$. Let $v \sim w$ denote the adjacency relation of G , and $\lambda : V(G) \rightarrow \{XY, YZ, XZ\}$ indicate the plane of measurement for each qubit so that for each qubit, $\lambda(u) = P$ only if $u \in M_P$. Then for any family $\{\theta_u\}_{u \in O^c}$ of measurement angles satisfying*

$$\left\{ \begin{array}{l} u \in XZ \cap YZ, \text{ or} \\ u \in XY \cap XZ \text{ and } \lambda(u) = XY \end{array} \right\} \implies \theta_u \in \{0, \pi\}, \quad (4.14a)$$

$$\left\{ \begin{array}{l} u \in XY \cap YZ, \text{ or} \\ u \in XY \cap XZ \text{ and } \lambda(u) = XZ \end{array} \right\} \implies \theta_u \in \{-\pi/2, \pi/2\}, \quad (4.14b)$$

and for any linear order \leq extending \preceq , the measurement procedure

$$\left[\prod_{u \in O^c}^{\leq} \left(\prod_{\substack{w \in \text{odd}(g(u)) \\ w \neq u}} Z_w^{s[u]} \right) \left(\prod_{\substack{v \in g(u) \\ v \neq u}} X_v^{s[u]} \right) M_u^{(\lambda(u), \theta_u)} \right] E_G N_{I^c} \quad (4.15)$$

performs a unitary transformation.

This represents a further extension of the possible range of 1WQC procedures which may be identified as performing unitary transformations; the proof of the above

^yWe may recover the definition of Pauli flows found in [4] by defining a function λ from O^c to $\{X, Y, Z, XY, YZ, XZ\}$, such that

- $\lambda(v) = X$ if $v \in M_{XY} \cap M_{XZ}$,
- $\lambda(v) = Y$ if $v \in M_{XY} \cap M_{YZ}$,
- $\lambda(v) = Z$ if $v \in M_{XZ} \cap M_{YZ}$, and
- $\lambda(v) = P$ if $v \in M_P$ otherwise;

the fact that $O^c = M_{XY} \cup M_{YZ} \cup M_{XZ}$ then implies that the conditions of Definition 4.2 and Definition 5 of [4] are equivalent.

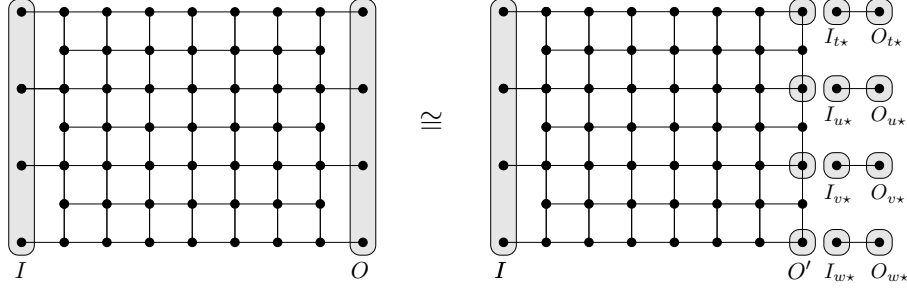


Figure 13: Illustration of the geometry for a qubit-reversal procedure of Section IV.B of [2], and a partial decomposition into star geometries. Each qubit in $V(G) \setminus O$ is to be measured in the XY plane with default angle 0: when this yields the measurement result $s[u] = 0$ for each $u \in V(G) \setminus O$, the result is to perform a reversal of the vertical ordering of the logical qubits: input states $|\psi\rangle_{a,b,c,d}$ of the input subsystem (with $a-d$ ordered from top to bottom) are mapped to $|\psi\rangle_{w,v,u,t}$ of the output subsystem (with $t-w$ ordered from top to bottom). Notice that the residual geometry in the decomposition on the right has no maximal star geometries, as every element of O' is adjacent to more than one element of $V(G) \setminus O'$, and every element of $V(G) \setminus O'$ is adjacent to at least one other element of $V(G) \setminus O'$. Thus, this geometry has no star decomposition, and therefore no modified flow.

Lemma may again be obtained by considering how each measurement may be certified to transform the state-space unitarily conditioned on a preferred measurement result, where the transformation of the state-space and the corrections to apply are described by the stabilizer formalism.

In order to use this to provide semantics for such 1WQC procedures in terms of unitary circuits, we may consider how the concept of star geometries generalize from extended flows to Pauli flows. The natural approach, rather than attempt to retain the star-graph structure, would be to consider geometries with a single root u , multiple “centers” $v \in g(u)$, and whose output subsystem consists of the qubits $[g(u) \cup \text{odd}(g(u))] \setminus \{u\}$. However, it is not as clear what the circuit corresponding to such a geometry would be, nor what results may be shown for the decomposition of a geometry (G, I, O) into such geometries, as these lack the simple local structure of star geometries. How a Pauli flow structure may be exploited to produce more general semantic maps for 1WQC procedures in unitary circuit models is an open question.

4.2 A simple 1WQC construction lacking a star decomposition

Lemma 4.2 above presents a 1WQC procedure which performs a unitary embedding, but for which the techniques of Section 3 do not apply. Another example is the procedure for performing reversals of a linear array of logical qubits presented in Section IV.A of [2]. Figure 13 illustrates the geometry underlying this 1WQC procedure: the entire procedure may be obtained from the specification that each qubit in O^c is

to be measured in the XY plane with the default angle 0, and noting that as this represents selection of the $+1$ eigenstate of the observables X_u for $u \in O^c$, from which suitable correction operations may be obtained by the stabilizer formalism. However, as Figure 13 also illustrates, this procedure has no star decomposition, and so cannot be mapped to its meaning as a reversal of a linear array of qubits using the techniques of Section 3.

Despite this, we may gain some information about the qubit-reversal pattern using modified flows by an appropriate extension of the input and output systems, as exhibited in [11]. It is easy to see that if we include the left-most element of each row into I , and the right-most into O , the resulting altered geometry has a flow with a successor function which maps each vertex to the one immediately to the right. Chains in the natural pre-order \preceq have monotonically increasing column numbers: and therefore \preceq is antisymmetric. Given that the default measurement angle for each qubit is zero, we may obtain a candidate circuit for this procedure, as illustrated in Figure 14. The final circuit illustrated there can easily be described as a reversible classical computation: for seven input bits (v_0, \dots, v_6) , the computation performed by that circuit is

$$\begin{aligned} & (v_0, v_1, v_2, v_3, v_4, v_5, v_6) \\ & \longmapsto (v_6, v_4 \oplus v_5 \oplus v_6, v_4, v_2 \oplus v_3 \oplus v_4, v_2, v_0 \oplus v_1 \oplus v_2, v_0). \end{aligned} \quad (4.16)$$

If we apply this circuit instead to qubits, with the odd-indexed qubits initialized to the $|+\rangle$ state (*i.e.* a uniform superposition of the bit-values described above), those qubits remain in the $|+\rangle$ state, and the even-indexed qubits are reversed, as required.

We may then understand the qubit-reversal pattern in [2] as corresponding to a circuit which prepares auxiliary qubits in the $|+\rangle$ state, and subsequently removes them by an X observable measurement. The fact that we are able to obtain the circuit in this case is possible because the highly structured geometry makes it easier to guess which qubit measurements correspond to terminal measurements in a unitary circuit. Without exploiting this structure, however, there are no known approaches to *algorithmically* provide semantics for such a measurement pattern in terms of unitary circuits.

5 Review and open problems

In this article, we have presented an algorithm for obtaining the “meaning” of 1WQC procedures \mathfrak{P} in terms of unitary circuit models, using a combinatorial analysis of the structures underlying procedure \mathfrak{P} arising from standard constructions of 1WQC procedures. In doing so, we have identified a correspondence between elementary combinatorial structures (*star geometries*) and elementary circuits (*star circuits*) which, given a 1WQC procedure whose geometry which may be decomposed into these elementary structures, allows the construction of a corresponding circuit. Together with a characterization of the dependencies of operations on classical measurement outcomes, this enables us to efficiently construct a circuit C for a 1WQC procedure \mathfrak{P} , and to determine whether C and \mathfrak{P} perform the same operation, when \mathfrak{P} has input and output subsystems of the same size.

We have used the gate model $\{H, T, \wedge Z\}$ as a reference model of circuits, using the related gate-sets $\mathcal{B}_{\text{DKP}} = \{J(\theta), \wedge Z\}_{\theta \in \frac{\pi}{4}\mathbb{Z}}$ and $\mathcal{B}_{\text{RBB}} = \{J(\theta), \mathcal{Z}z\}_{\theta \in \frac{\pi}{4}\mathbb{Z}}$ to de-

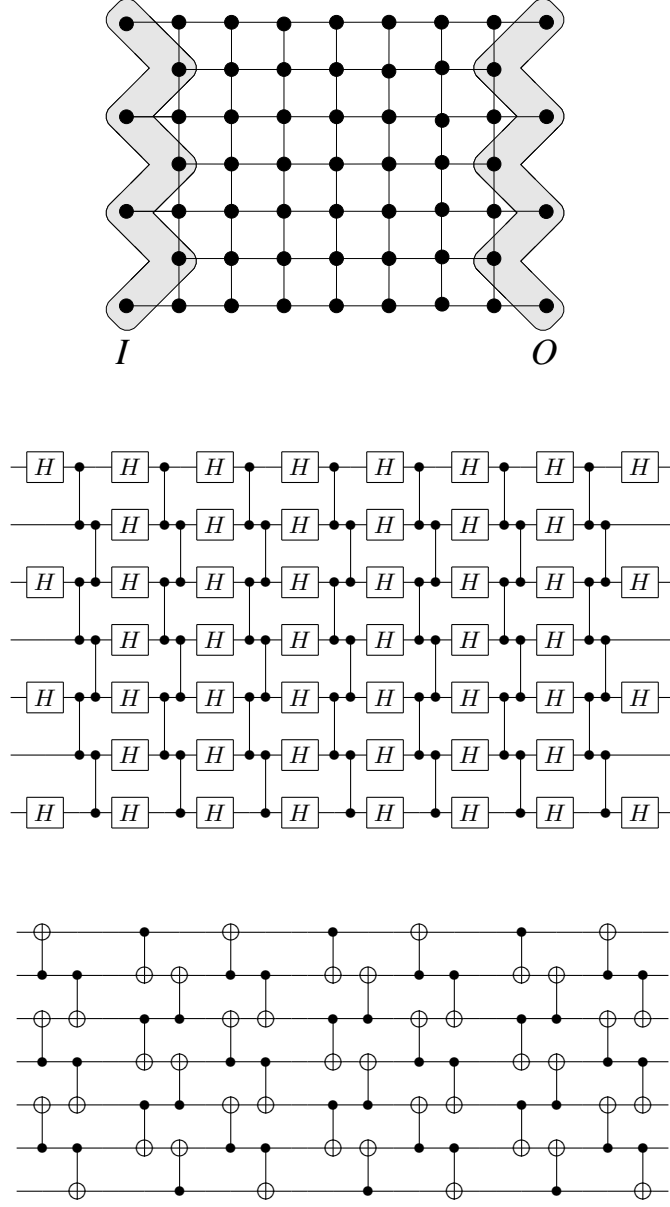


Figure 14: Illustration of the qubit-reversal procedure of Section IV.B of [2], with augmented input/output subsystems, and corresponding candidate circuits (represented in the middle in the gate-set $\{H, T, \wedge Z\}$, and in the bottom with CNOT gates).

scribe constructions of 1WQC procedures. However, essentially all of the results of this article generalize to $\{H, R(\theta), \wedge Z\}_{\theta \in \mathbb{R}}$, where

$$R(\theta) = e^{-i\theta Z/2} = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} : \quad (5.1)$$

we then have $J(\theta) = HR(\theta)$, and the gate sets $\{H, R(\theta), \wedge Z\}_{\theta \in \mathbb{R}}$, $\{J(\theta), \wedge Z\}_{\theta \in \mathbb{R}}$, and $\{J(\theta), \mathcal{Z}_z\}_{\theta \in \mathbb{R}}$ are all parsimonious in the sense of Definition 2.1.

There are three natural open problems: how this analysis may be extended to the case of unitary embeddings in general, how this may be extended to admit an analysis for quantum circuits with ancillas (qubits which are prepared, operated on, and subsequently removed without creating mixtures of states in the other qubits), and how we may extend this analysis to include stabilizer certificates such as Pauli flows.

General unitary embeddings. The problem presented by the case of a unitary embedding with $|I| < |O|$ is that a star decomposition of the geometry may not be unique. Consequently, there may be many modified flow functions against which the dependencies of a 1WQC procedure must in principle be tested. We may relieve the situation if we can use the dependencies themselves to narrow the search space for a single modified flow; this seems a likely approach to extending the analysis of this article to a semantic map for the DKP and (simplified) RBB constructions, without further constraints.

Quantum circuits with ancillas. The qubit-reversal procedure described in Section 4.2 presents a situation where the introduction and removal of ancillas can be used to obtain a unitary circuit which corresponds to a 1WQC procedure. Combinatorial tools for the “detection” of instances in which a qubit may represent the state of an ancilla qubit would yield more general tools for presenting the semantics of 1WQC procedures in terms of unitary circuits, possibly permitting a semantic map to be defined for a construction employing all of the sub-procedures described in Section IV of [2], and for 1WQC procedures in general arising from circuits involving partial trace operations on qubits which are disentangled from the rest of the system.

Extension to more general stabilizer certificates. As noted in Section 4.1.1 and Section 4.1.2, modified flows essentially represent a means via the stabilizer formalism to certify that, for a given geometry (G, I, O) , there is an efficiently describable 1WQC procedure having that geometry (G, I, O) and which performs a unitary transformation. We can then consider how to compare an arbitrary procedure with geometry (G, I, O) to this canonical geometry. In Section 4.1.1, we showed how such certificates may be extended to more general gate-sets: how we might exploit the definition of Pauli flows in Definition 4.2 to obtain more general semantic maps is an open problem.

Acknowledgements. This work is based substantially on results and remarks made in [9], which was written while the author was affiliated with the Institute for Quantum Computing at the University of Waterloo, and funded in part by MITACS and

DTO-ARO. The extension of the results to the simplified RBB construction was written during the author’s current affiliation with the Universität Potsdam, and was funded in part by EPSRC. Finally, the author would like to thank Ashwin Nayak, Andrew Childs, Michele Mosca, Debbie Leung, John Watrous, and Dan Browne for remarks on the presentation of [9].

References

- [1] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Physical Review Letters*, 86(5188), 2001.
- [2] R. Raussendorf, D. E. Browne, and H. J. Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2):022312, Aug 2003. [arXiv:quant-ph/0301052].
- [3] V. Danos, E. Kashefi, and P. Panangaden. Robust and parsimonious realisations of unitaries in the one-way model. *Physical Review A*, 72:064301, 2006.
- [4] D. E. Browne, E. Kashefi, M. Mhalla, and S. Perdrix. Generalized flow and determinism in measurement-based quantum computation. *New Journal of Physics*, 9(8):250, 2007. [arXiv:quant-ph/0702212].
- [5] D. Gottesman. *Stabilizer codes and quantum error correction*. PhD in Physics, California Institute of Technology, 1200 East California Blvd, Pasadena CA, 91125, 1997. [arXiv:quant-ph/9705052].
- [6] D. Gottesman and I. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402, 1999. Preprint version, entitled “Quantum Teleportation is a Universal Computational Primitive” available as [arXiv:quant-ph/9908010v1].
- [7] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. Pre-print: [arXiv:quant-ph/0001106], 2000.
- [8] E. Farhi, J. Goldstone, and S. Gutmann. A quantum algorithm for the hamiltonian NAND tree. Pre-print: [arXiv:quant-ph/0702144], 2007.
- [9] J. R. N. de Beaudrap. *Theory of measurement-based quantum computing*. PhD in Mathematics, University of Waterloo, 2008. Available online at [hdl.handle.net/10012/4133] and as [arXiv:0812.2869].
- [10] V. Danos, E. Kashefi, and P. Panangaden. The measurement calculus. *J. ACM*, 54(2):8, 2007. [arXiv:0704.1263].
- [11] Andrew M. Childs, Debbie W. Leung, and Michael A. Nielsen. Unified derivations of measurement-based schemes for quantum computation. *Physical Review A*, 71(3):032318, 2005. [arXiv:quant-ph/0404132].

- [12] V. Danos and E. Kashefi. Determinism in the one-way model. *Physical Review A*, 74:052310, 2006. [arXiv:quant-ph/0506062].
- [13] N. de Beaudrap. Finding flows in the one-way measurement model. *Physical Review A*, 77:022328, 2008. [arXiv:quant-ph/0611284].
- [14] M. Mhalla and S. Perdrix. Finding optimal flows efficiently. In *Lecture Notes in Computer Science*, volume 5125, pages 857–868, 2008. [arXiv:0709.2670].
- [15] N. de Beaudrap and M. Pei. An extremal result for geometries in the one-way measurement model. *Quantum Information & Computation*, 8(5):430–437, 2008. [arXiv:quant-ph/0702229].
- [16] R. Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, New York, 3rd edition, 2005. Electronic version freely available on the author’s website, at [www.math.uni-hamburg.de/home/diestel/books/graph.theory].
- [17] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000. See errata [20].
- [18] A. Einstein. The foundation of the general theory of relativity. *Annalen Phys.*, 49:769–822, 1916.
- [19] D. E. Browne and H. J. Briegel. One-way quantum computation — a tutorial introduction, 2006. [arXiv:quant-ph/0603226].
- [20] M. A. Nielsen and I. L. Chuang. Errata list for *Quantum Computation and Quantum Information*. Available online at [web.squint.org/qci/errata/errata/errata.html]; see [17].
- [21] C. M. Dawson, H. L. Haselgrove, A. P. Hines, D. Mortimer, M. A. Nielsen, , and T. J. Osborne. Quantum computing and polynomial equations over \mathbb{Z}_2 . *Quantum Information & Computation*, 5(2):102, 2004. [arXiv:quant-ph/0408129].

A Stable-index tensor notation

A.1 Sum-over-paths interpretation of stable-index tensors

The informal semantics of stable-index tensor notation is to provide a simplified description of circuits in terms of computational paths, as follows.

Stable-index notation for circuits is partially inspired by the *path-labelling* of circuits by Dawson *et al.* [21], which is explicitly concerned with descriptions of circuits in terms of computational paths: the tensor indices of the stable-index expressions correspond to labels of “wire segments” of [21], which are separated by Hadamard gates (or more generally, other operations which do not preserve the standard basis). For a given operator, a stable index s then represents a qubit for which the standard basis states $|0\rangle$ and $|1\rangle$ remain unchanged (*i.e.* the density operators $|s\rangle\langle s|$ are unaffected) by the operation for $s \in \{0, 1\}$. Over a single “wire-segment”, the reduced state of such a

qubit will then remain unchanged, even if one measures this state in the standard basis at the beginning of the wire-segment.

As with tensors in Einstein notation, we may describe a “path” as a mapping of each index (whether bound or free) to some fixed value; the amplitude associated with a path is the product of the corresponding coefficient of each factor involved in the expression. The convention of summing over bound indices (whose possible values may be thought to represent intermediate configurations which were not observed, and therefore not distinguished from one another) then represents a summation over indistinguishable paths. The simplification introduced by stable-index expressions corresponds to explicit recognition when operations leave some of the “dynamical variables” associated to a path (represented by tensor indices) unaffected.

A.2 Constructing stable-index representations

We may automatically generate a stable-index representation S for a unitary circuit from a representation as a composition of operators C on specified qubits in an efficient manner. We consider two distinct methods: one which is appropriate for a network without any topological constraints on the operators, and one which is motivated by the constraints described in Section 2.3.3.

A.2.1 Construction for topologically unconstrained circuits

Let L be the set of qubit labels in C . We may then transliterate the terms in C , defining a set of tensor indices incrementally as we do so, as follows:

1. For each qubit $v \in C$, we define the tensor index v_0 . We designate v_0 as *the current tensor index* for v . (Throughout the transliteration of the circuit, the index designated as “current” for a qubit will be updated.) We also start with an expression S consisting of the null sequence of operators: this may be regarded as representing a stable-index representation for the empty product (the identity operator).
2. We translate each operator $U_{a,b,\dots}$ in C taken in sequence, as follows. For each qubit $v \in \{a, b, \dots\}$ on which U acts (again, in sequence), let v_j be the current index for v , and perform the following:
 - (i) If $U_{a,b,\dots}$ introduces v as a “fresh” qubit (or a new qubit which is possibly entangled with the others), then v_0 is still the current index for v . We then use v_0 as the advanced index corresponding to v , without a corresponding deprecated index.
 - (ii) If $U_{a,b,\dots}$ not preserve each of the standard basis states on v (that is, if $U_{a,b,\dots} |s\rangle\langle s|_v U_{a,b,\dots}^\dagger \neq |s\rangle\langle s|_v$ for some $s \in \{0, 1\}$), then we define a new tensor index v_{j+1} for v . We then use v_j as the deprecated index corresponding to v , and v_{j+1} as the corresponding advanced index. We update the current index for v to v_{j+1} .

- (iii) If $U_{a,b,\dots}$ preserves each of the standard basis states on v (that is, for each $s \in \{0, 1\}$ we have $U_{a,b,\dots} |s\rangle\langle s|_v U_{a,b,\dots}^\dagger = |s\rangle\langle s|_v$), then we use v_j as a stable index corresponding to v , without defining any new indices.^z
- (iv) If $U_{a,b,\dots}$ removes a qubit v (e.g. describing an operation conditioned on a particular measurement outcome), then we use v_j as a deprecated index corresponding to v , without a corresponding advanced index.

This produces sequences of stable indices \mathbf{s} and sequences of matched advanced and deprecated indices \mathbf{a} and \mathbf{d} (possibly with placeholders); we then translate $U_{a,b,\dots}$ as $U[\mathbf{s} \atop \mathbf{a} \atop \mathbf{d}]$, and append it to S .

3. Having translated each of the terms in C , the sequence S is a stable-index tensor expression corresponding to C .

Note that in particular, every new index which is advanced by a gate $U[\mathbf{s} \atop \mathbf{a} \atop \mathbf{d}]$ must be distinct from any other index being advanced, as well as from all of the indices which have occurred to that point. This ensures that the resulting product is acyclic if the original network C was acyclic.

Note that the index successor function f (Definition 2.3) can be easily described from this construction: f has as a domain all those indices v_j such that v_{j+1} is well-defined, and $v_{j+1} = f(v_j)$ for all such indices.

A.2.2 Construction suitable for use in the simplified RBB construction

Motivated by the constraints described in Section 2.3.3 on 1WQC procedures whose entanglement graph is an induced subgraph of the grid, we also present a construction for stable-index expressions for circuits with a linear-nearest neighbor topology with the gate set $\mathcal{B}_{\text{RBB}} = \{J(\theta), \mathcal{Z}\mathcal{Z}\}_{\theta \in \frac{\pi}{4}\mathbb{Z}}$, in which the two-qubit gates

- (i) act “stably” on both their indices,
- (ii) can only act on indices of equal “depth” from the input — that is, can only act on pairs of indices v_j and w_j , where $v_j = f^j(v_0)$ and $w_j = f^j(w_0)$ for some indices $v_0, w_0 \notin \text{dom}(f)$, where f is the successor function of the stable-index expression; and
- (iii) can only act on a pair of indices v_j, w_j if there is no two-qubit gate acting on v_{j-1} and w_{j-1} .

Given a circuit C using the gates of \mathcal{B}_{RBB} , we may produce such a stable-index representation (and at the same time explicitly construct its interaction graph, applying Definition 2.4 in the special case of interactions of at most two qubits) as follows, adapting the algorithm of Section A.2.1 above.

^zIn the definition of stable-index notation in Section 2.1.3, all stable indices must occur as a contiguous block at the beginning of the linear ordering of the qubits operated on. Generalizing this definition, we may easily alternate between sequences of stable indices and sequences of advanced and deprecated indices. If we choose not to generalize stable-index notation in this way, we may only represent a qubit v with a stable index if all preceding qubits on which U operates are also represented by stable indices; otherwise, we must apply case (ii) above regardless of how U acts on v .

1. For each qubit v acted on by C , we set the current index to v_0 . We also initialize the graph G to the totally disconnected graph whose vertices are the indices v_0 . We start with an expression S consisting of the null sequence of operators: this may be regarded as representing a stable-index representation for the empty product (the identity operator).
2. We translate each operator $J(\theta)_a$ or \mathcal{Z}_{ab} in C taken in sequence, as follows. For each qubit v on which the gate acts, let v_j be the current index for v . For a gate $J(\theta)$, we append $J(\theta)\left[\frac{v_{j+1}}{v_j}\right]$ to S , add the index v_{j+1} and the edge $v_j v_{j+1}$ to G , and set the current index to v_{j+1} . Otherwise, for a gate \mathcal{Z}_{vw} , we perform the following operations:

- (i) We ensure that the constraints on successive applications of two-qubit gates is satisfied as follows. If the current indices for v and w are v_j and w_j for a common value of j , and there is an edge of the form $v_{j-1} w_{j-1}$ in G , we append the expression

$$J(0)\left[\frac{v_{j+2}}{v_{j+1}}\right] J(0)\left[\frac{v_{j+1}}{v_j}\right] J(0)\left[\frac{w_{k+2}}{w_{k+1}}\right] J(0)\left[\frac{w_{k+1}}{w_k}\right] \quad (\text{A.1})$$

to S , and add the edges $v_j v_{j+1}$, $v_{j+1} v_{j+2}$, $w_j w_{j+1}$, and $w_{j+1} w_{j+2}$ (along with the indicated indices) to G . We then set the current indices of v and w to v_{j+2} and w_{j+2} respectively.

- (ii) We then recursively perform the following, letting v_j and w_k stand for the current indices of v and w at each iteration:

- If $|k - j| \geq 3$ or $|k - j| = 2$, we append either

$$J(0)\left[\frac{v_{j+2}}{v_{j+1}}\right] J(0)\left[\frac{v_{j+1}}{v_j}\right] \quad \text{or} \quad J(0)\left[\frac{w_{k+2}}{w_{k+1}}\right] J(0)\left[\frac{w_{k+1}}{w_k}\right] \quad (\text{A.2})$$

to S , according to whether $k > j$ or $j > k$ respectively, adding the appropriate indices and edges to G . We set the current index either of v or w to v_{j+2} or w_{k+2} accordingly, and repeat for the new current indices.

- If $|k - j| = 3$ or $|k - j| = 1$, we append either

$$\begin{aligned} & J(\pi/2)\left[\frac{v_{j+3}}{v_{j+2}}\right] J(\pi/2)\left[\frac{v_{j+2}}{v_{j+1}}\right] J(\pi/2)\left[\frac{v_{j+1}}{v_j}\right] \quad \text{or} \\ & J(\pi/2)\left[\frac{w_{k+3}}{w_{k+2}}\right] J(\pi/2)\left[\frac{w_{k+2}}{w_{k+1}}\right] J(\pi/2)\left[\frac{w_{k+1}}{w_k}\right] \end{aligned} \quad (\text{A.3})$$

to S , according to whether $k > j$ or $j > k$ respectively, adding the appropriate indices and edges to G . We set the current index either of v or w to v_{j+3} or w_{k+3} accordingly, and repeat for the new current indices.

- If $j = k$, we append $\mathcal{Z}_{[v_j, w_j]}$ to S , add the edge $v_j w_j$ to G , and stop.

3. Having translated each of the terms in C , the sequence S is a stable-index tensor expression corresponding to C , and G its interaction graph.

In the above construction, note that the final index v_ℓ for each qubit v may not have the same value of ℓ ; and that such stable-index expressions may both begin and end with operations $\mathcal{Zz}[v_j, w_j]$. These degrees of freedom may complicate the procedure of composing two such circuits. These issues may both be mitigated by appending appropriate products of $J(\theta)$ operators which evaluate to the identity, of differing lengths for each qubit; this can be done to both uniformize the value ℓ of the final indices v_ℓ , and also ensure that any \mathcal{Zz} operation is a distance of at least 2 from the end of the circuit. However, for the purposes of this article, such operations will not be necessary, as we do not consider the composition of two such expressions.

A.3 Congruence of circuits and isomorphic stable-index expressions

In this section, we prove Theorem 2.1 (page 11), and expand on the role of parsimonious sets of gates in this result.

Lemma A.1. *Let C_1 and C_2 be two sequences of unitary operators on a common set of qubits, composed from a common set of unitaries. If the stable[index tensor representations of C_1 and C_2 are isomorphic, then C_1 and C_2 are congruent up to re-ordering of commuting operators.*

Proof. Consider the coarsest equivalence relation \cong on circuits which consist of permutations of the gates of C_1 , such that $C \cong C'$ if C differs from C' by a transposition of commuting gates. By definition, the equivalence classes of \cong are sets of circuits which are congruent up to re-ordering of commuting operators. Consider stable-index tensor expressions S and S' arising from two unitary circuits C and C' , where S acts on the same index set as S' , and S differs from S' only by a transposition of two terms $U_1[s_1 \frac{a_1}{d_1}]$ and $U_2[s_2 \frac{a_2}{d_2}]$, where without loss of generality the former precedes the latter in C , and vice-versa in C' . Then, the corresponding gates U_1 and U_2 in C occur with U_1 preceding U_2 , and occur in C' in the opposite order. Consider the index sets for these two operations in the stable-index tensor expression:

- If the sequences of indices $(s_1; a_1; d_1)$ and $(s_2; a_2; d_2)$ do not overlap, the operations U_1 and U_2 act on disjoint sets of qubits, and so they commute.
- Suppose the sequences of indices $(s_1; a_1; d_1)$ and $(s_2; a_2; d_2)$ overlap, and let v be an index occurring in both sequences. In the construction of S , because v occurs in $U_2[s_2 \frac{a_2}{d_2}]$, it cannot be a deprecated index in $U_1[s_1 \frac{a_1}{d_1}]$; and because it occurs in $U_1[s_1 \frac{a_1}{d_1}]$, it cannot be an advanced index in $U_2[s_2 \frac{a_2}{d_2}]$. By the construction of S' , we similarly have that v cannot be a deprecated index in $U_2[s_2 \frac{a_2}{d_2}]$ or an advanced index in $U_1[s_1 \frac{a_1}{d_1}]$. Then, v is a stable index of both terms.

Therefore, the only qubits v on which U_1 and U_2 both act are those whose standard basis states are preserved by both U_1 and U_2 . Then, U_1 and U_2 commute, in which case $C \cong C'$.

Let S_1 and S_2 be the stable-index tensor representations of C_1 and C_2 , where the isomorphism is given by an index relabelling map λ and mapping of the terms τ . Let

S'_2 be the representation obtained by applying λ to all of the indices in S_2 : then S_1 and S'_2 consist of a product of the same terms in different orders. Without loss of generality, we will then suppose that S_2 and S_1 differ only by a reordering of terms. Because the terms of S_1 differ from those of S_2 by a permutation, by induction on the decomposition of this permutation into transpositions, we therefore have $C_1 \cong C_2$. ■

Whatever set of gates is used in a stable-index tensor expressions, rearranging the terms in the expression not only preserves the meaning (by summation over indices which are both advanced and deprecated) as a unitary operator, but also preserves the relevant structural information about the circuit — preserving the necessary order information in the way described following Definition 2.3, while discarding such information for each pair of commuting operators. As a consequence, we may regard permuting terms of such an expression as a way of exploring the elements of the congruence class of a circuit.

However, in the case where two gates commute without preserving the standard basis states of their common operands, we may obtain congruent circuits which do not have isomorphic tensor index expansions:

Example 5. Let $U \in \text{U}(2)$ be a non-diagonal operator, and define the operation $\wedge U$ as in (2.12a). Consider the two circuits $\wedge U_{c,b} \wedge U_{a,b} H_a$ and $\wedge U_{a,b} \wedge U_{c,b} H_a$ on three qubits a, b, c (where the subscripts here denote the linear order of the qubits as operands of $\wedge U$ and H): these have the respective stable-index expressions

$$\wedge U \left[c \begin{smallmatrix} b_3 \\ b_2 \end{smallmatrix} \right] \wedge U \left[a_1 \begin{smallmatrix} b_2 \\ b_1 \end{smallmatrix} \right] H \left[\begin{smallmatrix} a_1 \\ a_0 \end{smallmatrix} \right] \quad \text{and} \quad \wedge U \left[a_1 \begin{smallmatrix} b_3 \\ b_2 \end{smallmatrix} \right] \wedge U \left[c \begin{smallmatrix} b_2 \\ b_1 \end{smallmatrix} \right] H \left[\begin{smallmatrix} a_1 \\ a_0 \end{smallmatrix} \right]. \quad (\text{A.4})$$

These two expressions are non-isomorphic: in the first expression, the leftmost factor contains the indices c (which is neither advanced nor deprecated in the entire expression) and b_3 (which is not deprecated in the entire expression); the second expression does not contain any factor with these properties.

In the example above, the inequivalence of the two stable-index expressions is due to the fact that the two operations $\wedge U_{a,b}$ and $\wedge U_{b,c}$ preserve the eigenbasis of U on b , rather than the basis $|0\rangle, |1\rangle$. The restrictions imposed on parsimonious sets of unitary operations prevent precisely this problem from arising, which allows us to prove the following:

Lemma A.2. Let C_1 and C_2 be two unitary circuits composed from a common parsimonious set of unitaries. If C_1 is congruent to C_2 up to re-ordering of commuting operators, then the stable-index representations of C_1 and C_2 will be isomorphic.

Proof. Let $K_0 \cong K_1 \cong \dots \cong K_\ell$ be a sequence of circuits which differ only by a transposition of two commuting gates, where $C_1 = K_0$ and $C_2 = K_\ell$. We may show by induction on ℓ that the stable-index tensor expressions for C_1 and C_2 are isomorphic; it suffices in this case to prove the case $\ell = 1$.

Let S_0 and S_1 be the stable-index representations of K_0 and K_1 . Up to a relabeling, we may assume that the indices of S_0 and S_1 are all of the form v_0, v_1, \dots for different qubits v acted on by K_0 and K_1 : we set v_0 to be the first index corresponding to a given qubit in either S_0 or S_1 , and v_{j+1} to be an advanced index corresponding to each

deprecated index v_j for $j \in \mathbb{N}$. Let U_1 and U_2 be the pair of gates whose order differs between the two circuits; and consider the truncation S'_0 of S_0 just prior to the terms corresponding to U_1 and U_2 , and similarly for S'_1 . Because K_0 and K_1 differ only by a transposition of commuting gates U_1 and U_2 , $S'_0 = S'_1$. Let S''_0 and S''_1 be the truncations of S_0 and S_1 just after the terms corresponding to U_1 and U_2 :

- If U_1 and U_2 act on disjoint sets of qubits, the terms corresponding to U_1 and U_2 in S''_0 may differ from those in S''_1 only by the indices which are advanced in each; and as the indexing scheme we have fixed is the same for each, and the indices which will be advanced for U_1 and U_2 are independent of each other in both expressions, S''_0 differs from S''_1 only by a transposition of those two terms.
- If U_1 and U_2 both have the standard basis as their eigenbases, their corresponding terms in S_0 and S_1 only have stable indices, and therefore do not advance any new indices. Then the term in S_0 and in S_1 corresponding to U_1 are the same, and similarly for U_2 ; the difference between S''_0 and S''_1 is a transposition of those two terms, and so they are isomorphic.
- If U_1 and U_2 perform the same single-qubit operation, there will be some common tensor index u_j which is current for that qubit prior to the application of U_1 and U_2 in both S'_0 and S'_1 . Then, the terms corresponding to U_1 and U_2 in S_0 will be $U_2\left[\frac{u_{j+2}}{u_{j+1}}\right] U_1\left[\frac{u_{j+1}}{u_j}\right]$, and the corresponding terms in S_1 will be $U_1\left[\frac{u_{j+2}}{u_{j+1}}\right] U_2\left[\frac{u_{j+1}}{u_j}\right]$ for some indices v'' and w'' . Given that U_1 and U_2 perform the same operation, we then have $S''_0 = S''_1$.

Because the operations of K_0 and K_1 are identical after U_1 and U_2 , the differences between S_0 and S_1 after the terms corresponding to U_1 and U_2 can only arise from the indices advanced by each term; because we have fixed the indexing scheme, the only difference between S_0 and S_1 is then the (possibly trivial) transposition of those two terms. Thus, S_0 and S_1 are isomorphic. ■

Theorem 2.1 then follows from Lemma A.1 and A.2. Using this, we may identify congruent circuits simply by relabelling their indices in a uniform way (*e.g.* as in the construction for stable-index expressions in Section A.2), and determining whether there is a perfect matching between the two expressions defined by equality of terms.

B Details for constructions of 1WQC procedures

B.1 Construction of the gate-sets \mathcal{B}_{DKP} and \mathcal{B}_{RBB}

In this section, we provide details on the construction of the 1WQC procedures \mathfrak{J}^θ and \mathfrak{Z}_3 , as well as a generalization of the latter operation to a many-qubit diagonal operation.

B.1.1 Two-qubit operations

As described in (2.40), we may implement $\mathcal{Z}z$ in the 1WQC model with the procedure

$$\mathfrak{Z}\mathfrak{Z}_{v,w} = Z_v^{s[a]} Z_w^{s[a]} M_a^{\pi/2} E_{av} E_{aw} N_a . \quad (\text{B.1})$$

For the analysis of Section 3, it will be convenient to generalize this to a procedure $\mathfrak{Z}\mathfrak{Z}_{v_1, \dots, v_d}^d$ acting symmetrically on many qubits:

$$\mathfrak{Z}\mathfrak{Z}_{v_1, \dots, v_d}^d = \left(\prod_{j=1}^d Z_{v_j}^{s[a]} \right) M_a^{\pi/2} \left(\prod_{j=1}^d E_{av_j} \right) N_a . \quad (\text{B.2})$$

We may compute the effect of this procedures as follows. For standard basis vectors $|x_j\rangle, |x'_j\rangle$ for each $1 \leq j \leq d$, ignoring the measurement result $s[a]$ after the correction operations, we may obtain

$$\begin{aligned} & \mathfrak{Z}\mathfrak{Z}_{v_1, \dots, v_d}^d \left(\bigotimes_{j=1}^d |x_j\rangle\langle x'_j|_{v_j} \right) \\ &= \left(\prod_{j=1}^d Z_{v_j}^{s[a]} \right) M_a^{\pi/2} \left(\left[Z^{x_1+\dots+x_d} |+\rangle\langle +| Z^{x'_1+\dots+x'_d} \right]_a \otimes \left(\bigotimes_{j=1}^d |x_j\rangle\langle x'_j|_{v_j} \right) \right) \\ &= \sigma^+ \left(\bigotimes_{j=1}^d |x_j\rangle\langle x'_j|_{v_j} \right) + \sigma^- \left(\bigotimes_{j=1}^d \left[Z |x_j\rangle\langle x'_j| Z \right]_{v_j} \right), \end{aligned} \quad (\text{B.3a})$$

where we define the non-negative scalars σ^\pm by

$$\sigma^+ = \langle +_{\pi/2} | Z^{x_1+\dots+x_d} |+\rangle\langle +| Z^{x'_1+\dots+x'_d} |+\pi/2\rangle , \quad (\text{B.3b})$$

$$\sigma^- = \langle -_{\pi/2} | Z^{x_1+\dots+x_d} |+\rangle\langle +| Z^{x'_1+\dots+x'_d} |-\pi/2\rangle . \quad (\text{B.3c})$$

We may consider the terms of (B.3a) as outer products of standard-basis vectors: we may verify that

$$\begin{aligned} & \left[\langle +_{\pi/2} | Z^{x_1+\dots+x_d} |+\rangle \right] \otimes |x_1\rangle_{v_1} \otimes \dots \otimes |x_d\rangle_{v_d} \\ &= \frac{1}{2} \left[1 - i(-1)^{x_1+\dots+x_d} \right] |x_1 \dots x_d\rangle_{v_1 \dots v_d} \\ &= \frac{1}{\sqrt{2}} e^{i\pi[2(x_1+\dots+x_d)-1]/4} |x_1 \dots x_d\rangle_{v_1 \dots v_d} \\ &= \frac{1}{\sqrt{2}} \left[e^{i\pi[Z \otimes \dots \otimes Z]/4} |x_1 \dots x_d\rangle \right]_{v_1 \dots v_d} , \end{aligned} \quad (\text{B.4a})$$

$$\begin{aligned}
& \left[\langle -\pi/2 | Z^{x_1+\dots+x_d} | + \rangle \right] \otimes \left[Z_v |x_1\rangle_{v_1} \right] \otimes \dots \otimes \left[Z_w |x_d\rangle_{v_d} \right] \\
&= \frac{1}{2} [(-1)^{x_1+\dots+x_d} + i] |x_1 \dots x_d\rangle_{v_1 \dots v_d} \\
&= \frac{i}{2} [1 - i(-1)^{x_1+\dots+x_d}] |x_1 \dots x_d\rangle_{v_1 \dots v_d} \\
&= \frac{i}{\sqrt{2}} \left[e^{i\pi[Z \otimes \dots \otimes Z]/4} |x_1 \dots x_d\rangle \right]_{v_1 \dots v_d} . \quad (\text{B.4b})
\end{aligned}$$

By taking outer products, and extending (B.3a) linearly over tensor products of the rank-1 projectors $|x_1\rangle\langle x'_1| \otimes \dots \otimes |x_d\rangle\langle x'_d|$ with linear operators on an external system S' , we then obtain

$$\mathfrak{Z}_{v_1, \dots, v_d}^d(\rho_S) = e^{-i\pi[Z_{v_1} \otimes \dots \otimes Z_{v_d}]/4} \rho_S e^{i\pi[Z_{v_1} \otimes \dots \otimes Z_{v_d}]/4} , \quad (\text{B.5})$$

for arbitrary operators ρ_S , acting on a system S which includes the qubits v_j but not the qubit a . In the special case $d = 2$, we then recover

$$\mathfrak{Z}_{v,w}(\rho_S) = e^{-i\pi[Z_v \otimes Z_w]/4} \rho_S e^{i\pi[Z_v \otimes Z_w]/4} = \mathcal{Z}_{z_{vw}} \rho_S \mathcal{Z}_{z_{vw}} , \quad (\text{B.6})$$

as claimed in (2.42).

B.1.2 Single qubit operations

We may describe 1WQC procedures for the operators $J(\theta)$ with a similar analysis as for the $\mathcal{Z}z$ operator above. For an arbitrary angle $-\pi < \theta \leq \pi$ and qubits v and w , define the CPTP map $\mathfrak{J}_{w/v}^\theta$ by

$$\mathfrak{J}_{w/v}^\theta(\rho_S) = X_w^{s[v]} M_v^{-\theta} E_{vw} N_w , \quad (\text{B.7})$$

for ρ a state supported on a set of qubits S which includes v but not w . We may characterize the effect of $\mathfrak{J}_{w/v}^\theta$ as follows. For $|\psi\rangle \in \mathcal{H}_2$ arbitrary, we have

$$\mathfrak{J}_{w/v}^\theta(|\psi\rangle\langle\psi|_v) = X_w^{s[v]} M_v^{-\theta} \left[\wedge_{Z_{vw}} \left(|\psi\rangle\langle\psi|_v \otimes |+\rangle\langle+|_w \right) \wedge_{Z_{vw}} \right] . \quad (\text{B.8})$$

Ignoring the measurement result $s[v]$ after the correction operation, we may characterize the compound operation $X_w^{s[v]} M_v^\varphi$ on a joint pure state $|\Psi\rangle_{v,w}$ by

$$\begin{aligned}
X_w^{s[v]} M_v^\varphi \left(|\Psi\rangle\langle\Psi|_{v,w} \right) &= \left(\langle +_\varphi |_v |\Psi\rangle_{v,w} \right) \left(\langle \Psi |_{v,w} | +_\varphi \rangle_v \right) \\
&\quad + \left(X_w \langle -_\varphi |_v |\Psi\rangle_{v,w} \right) \left(\langle \Psi |_{v,w} | -_\varphi \rangle_v X_w \right) . \quad (\text{B.9})
\end{aligned}$$

In the case $|\Psi\rangle_{v,w} = \wedge Z_{v,w} |\psi\rangle_v |+\rangle_w$, the terms in this equation may again be expressed as outer products of state vectors:

$$\begin{aligned} \langle +\varphi |_v \left[\wedge Z_{v,w} |\psi\rangle_v |+\rangle_w \right] &= \left[\langle +\varphi |_v \wedge Z_{v,w} |+\varphi\rangle_w \right] |\psi\rangle_v \\ &= J(-\varphi)_{w/v} |\psi\rangle_v, \end{aligned} \quad (\text{B.10a})$$

$$\begin{aligned} X_w \langle -\varphi |_v \left[\wedge Z_{v,w} |\psi\rangle_v |+\rangle_w \right] &= \left[X_w \langle -\varphi |_v \wedge Z_{v,w} |+\varphi\rangle_w \right] |\psi\rangle_v \\ &= \left[\langle +\varphi |_v \wedge Z_{v,w} |+\varphi\rangle_w \right] |\psi\rangle_v \\ &= J(-\varphi)_{w/v} |\psi\rangle_v, \end{aligned} \quad (\text{B.10b})$$

where by $J(-\varphi)_{w/v}$ we denote the linear operator $\mathcal{H}_v \rightarrow \mathcal{H}_w$ which maps state-vectors $|\psi\rangle_v \mapsto [J(-\varphi) |\psi\rangle]_w$. By linearity, we may extend this to arbitrary entangled states $|\psi\rangle_v \otimes |\phi\rangle_{S'}$ with a system S' which does not include v . Then, for a system S which includes v and excludes w , we have

$$\begin{aligned} \mathfrak{J}_{w/v}^\theta \left(|\Psi\rangle \langle \Psi|_S \right) &= \mathbf{X}_w^{s[v]} \mathbf{M}_v^{-\theta} \left[\wedge Z_{vw} \left(|\Psi\rangle \langle \Psi|_S \otimes |+\rangle \langle +|_w \right) \wedge Z_{vw} \right] \\ &= J(\theta)_{w/v} |\Psi\rangle \langle \Psi|_S J(\theta)_{v/w}^\dagger. \end{aligned} \quad (\text{B.11})$$

B.2 Efficiency of the DKP and simplified RBB constructions

Both the DKP construction and the (simplified) RBB construction, as described in Section 2.3.4, can be performed efficiently in the size of the input circuit. We may illustrate this for both procedures, as follows. In both cases, let k be the number of qubits which C acts upon, N be the number of one-qubit gates, and M be the number of two-qubit gates: we may assume that $k \leq N + M$ if we suppose that C acts non-trivially on each qubit.

B.2.1 Obtaining a circuit in normal form

We may easily construct a normal form in each case in a single pass of the expression of the circuit C . For each qubit v acted on by C , we may maintain a buffer B_v of single-qubit operations which are performed. These are initially empty for each qubit; we accumulate T and T^\dagger operations (cancelling them as appropriate) of each qubit as we encounter them; and upon encountering a Hadamard operation on a qubit v , we insert the contents of the buffer immediately before the Hadamard, resetting the buffer to zero after the Hadamard operation has been traversed. In both cases, we may also cancel $\wedge Z$ operations by storing in the buffers B_v indications of which qubits w the qubit v has interacted with via a $\wedge Z$: initially this is set to zero, and we toggle the presence of each qubit in the set with each $\wedge Z$ operation between the two qubits.

When we empty a buffer for a qubit v , we also produce the corresponding $\wedge Z$ gates between v and other qubits w . When we clear a buffer B_v prior to a Hadamard, we may count the number of $\wedge Z$ gates between v and each qubit w , and produce a $\wedge Z$ operation when the total is odd. For each $\wedge Z$ gate between v and some other qubit

w , we also remove a $\wedge Z$ gate in the buffer B_w between w and v to avoid double-counting the $\wedge Z$ gates. For the RBB construction, we may elaborate this further by converting $\wedge Z$ operations to $\mathcal{Z}z$ operations by performing the substitution $\wedge Z \mapsto \mathcal{Z}z(T^\dagger \otimes T^\dagger)^2$ when we encounter such an operation, accumulating the resulting T^\dagger gates in the buffers B_v and B_w when we do so.

The total amount of work performed by this routine per operation in C (counting both insertions and removals of terms corresponding to these operations, to or from the buffers) is $O(1)$. Using list-structures to implement the buffers B_v and adjacency-list representations for the $\wedge Z$ operations, we may initialize the buffers in time $O(k)$. The amount of work required by this phase is then $O(k + N + M)$.

B.2.2 Conversion to stable-index representation

This phase may be performed simultaneously with the previous phase: with each gate produced, we may produce a stable-index representation of the gate, and we produce $J(\theta)$ gates rather than H , T , and T^\dagger gates when we clear the buffers. In the case of the RBB construction, an additional amount of work may be required for the insertion of $J(\theta)$ gates to achieve uniform wire lengths: the amount of additional work per $\mathcal{Z}z$ gate is bounded to within a constant by the number of single-qubit operations performed on either wire, which is at most N . The amount of work required to do this is then $O(k + N + M)$ for the DKP construction, and $O(k + NM)$ for the RBB construction.

B.2.3 Translation of individual operations to the 1WQC model

This phase can also be performed simultaneously with the previous phases: rather than produce a stable-index expression C'' , we may produce the 1WQC procedure $\Phi(U[\dots])$ for each term $U[\dots]$ produced for the stable-index expansion. As each such 1WQC procedure is of constant length, this requires no additional overhead asymptotically.

B.2.4 Obtaining a 1WQC procedure in normal form

Let \mathfrak{P}_0 be the resulting 1WQC procedure from the previous phase. Let n be the number of qubits acted on by \mathfrak{P}_0 , and let m be the number of operations \mathfrak{J}^θ , E , or \mathfrak{Z}_3 in \mathfrak{P}_0 . (We may easily show that $n = k + N$ for the DKP construction and $n = k + N + M \in O(N + M)$ for the RBB construction; we have $m = N + M$ for the DKP construction and $m \leq N + NM \in O(NM)$ for the RBB construction, essentially by the arguments given for the run-time for the production of the stable-index expression C'' .) As each operation \mathfrak{J}^θ , E , and \mathfrak{Z}_3 consists of a constant number of elementary operations and contains only one or two entangling operations, the procedure \mathfrak{P}_0 also has $O(m)$ elementary operations and $O(m)$ entangling operations.

Standardizing \mathfrak{P}_0 . Let G be the entanglement graph associated to \mathfrak{P}_0 . In the process of standardizing \mathfrak{P}_0 , for each qubit v which is part of a $\mathfrak{J}_{v/u}$ operation for some u , we must commute a single operation $X_v^{S[u]}$ past entangling operations E_{vw} for all w adjacent

to v in the graph G . Rather than explicitly commuting operations, we may standardize \mathfrak{P}_0 by preparing a new 1WQC procedure \mathfrak{P}_1 as follows:

1. We initialize \mathfrak{P}_1 by assigning to it the procedure consisting of all preparation maps N_v in \mathfrak{P}_0 , followed by all entangling maps E_{vw} . (Note that the sets of prepared qubits and the graph G of entangling operators can be obtained with no additional asymptotic cost, during the phase of translation to the 1WQC model in which \mathfrak{P}_0 is produced.)
2. Note that every deprecated index v_j in the stable-index expansion of C'' has a corresponding advanced index v_{j+1} . As the procedures $\mathfrak{J}_{v/u}^\theta$ are the only operations in the DKP construction which allocate or discard qubits, each qubit u which is discarded in \mathfrak{P}_0 has a corresponding qubit v which is allocated, and this correspondence arises precisely from the presence of an operation $\mathfrak{J}_{v/u}^\theta$ in \mathfrak{P}_0 . If f is the index successor function of C'' , we then have $v = f(u)$ for such qubits u and v . For the RBB construction, we similarly have a gate $\mathfrak{J}_{v/u}^\theta$ for qubits with $f(u) = v$; but we also have qubits a which are allocated and discarded in the operations \mathfrak{J}_3 , which lie outside of the domain of f . We may then extend f to a function \tilde{f} such that $\tilde{f}(a) = a$ for such mediating qubits, and $\tilde{f}(v_j) = f(v_j) = v_{j+1}$ for $v_j \in \text{dom}(f)$. (Such a function \tilde{f} corresponds to an extended successor function as defined in Definition 3.1.) For the DKP construction, we will use $\tilde{f} = f$. Note that we may again construct \tilde{f} at no additional cost during the phase in which \mathfrak{P}_0 is produced from the stable-index expression C'' .

We may use the function \tilde{f} to append the measurements of \mathfrak{P}_0 to the left-hand side of \mathfrak{P}_1 in order, with the correct sign- and bit-dependencies included. We may make a single pass of \mathfrak{P}_0 , and in doing so maintain a list of correction operations for each of the qubits in \mathfrak{P}_0 , as follows. Initially, no qubit has any corrections. For any measurement M_u^* encountered, we perform the following:

- (i) If u does not have any accumulated corrections, we append M_u^θ to the left-hand side of \mathfrak{P}_1 without any dependencies;
- (ii) If u has accumulated corrections X_u^β and Z_u^γ , we append $S_{s[u]}^\gamma M_u^{\theta;\beta}$ to the left-hand side of \mathfrak{P}_1 . (If $\gamma = 0$, we may omit the shift operation.)
- (iii) We add an $X_v^{s[u]}$ correction to $v = \tilde{f}(u)$, provided that $v \neq u$, corresponding to the X correction arising from an operation $\mathfrak{J}_{v/u}^\theta$. (Note that the condition $u \neq \tilde{f}(u)$ is satisfied for all u in the DKP construction; it is satisfied if and only if $u \in \text{dom}(f)$ in the RBB construction.)
- (iv) We add $Z_w^{s[u]}$ corrections for every w adjacent to $v = \tilde{f}(u)$ in the graph G , provided $w \neq u$. (In the DKP construction, and for qubits $u \in \text{dom}(f)$ in the RBB construction, these corrections arise from commuting the $X_v^{s[u]}$ operation past the entangling maps E_{vw} for $w \neq u$; for qubits $a \notin \text{dom}(f)$ in the latter construction, this corresponds to the $Z_w^{s[a]}$ corrections in the \mathfrak{J}_3 procedure, in which case the condition $w \neq u$ will be satisfied for all w .)

We may simplify the expressions for any accumulated corrections using the relations $X_v^\beta X_v^{s[u]} = X_v^{\beta+s[u]}$ and $Z_w^\beta Z_w^{s[u]} = Z_w^{\beta+s[u]}$.

The amount of work performed for each measurement M_u^* in \mathfrak{P}_0 is at most $O(\deg[\tilde{f}(u)])$, where $\deg(v)$ is the number of neighbors of v in G . The total number of operations performed in this stage is then $O(\sum_v \deg(v)) = O(|E(G)|) = O(m)$.

3. After we have read all of the measurements, we append the corrections for each unmeasured qubit (computed in the previous step) to the end of \mathfrak{P}_1 .

The resulting procedure \mathfrak{P}_1 is then the same that would result from standardizing \mathfrak{P}_0 , and can be obtained in time $O(m)$.

Normalizing \mathfrak{P}_1 . Obtaining a normal form $\bar{\mathfrak{P}}$ from \mathfrak{P}_1 can be done by checking for Pauli simplifications for each measurement, and then performing signal shifting by accumulating shift operators to the left (without commuting shift operators past each other). We may check for and apply Pauli simplifications as a part of the standardization process above, with no additional overhead asymptotically. We may also perform signal shifting simultaneously with the standardization process, by immediately commuting the shift operation $S_{s[u]}^\gamma$ past any corrections $X_v^{s[u]}$ and $Z_w^{s[u]}$ which might have been introduced by the measurement on a given qubit u : if $v \sim w$ denotes the adjacency relation in G , we have

$$\begin{aligned} & \left(\prod_{\substack{w \sim \tilde{f}(v) \\ w \neq u}} Z_w^{s[u]} \right) \left(\prod_{\substack{v = \tilde{f}(v) \\ v \neq u}} X_v^{s[u]} \right) S_{s[u]}^\gamma \\ &= S_{s[u]}^\gamma \left(\prod_{\substack{w \sim \tilde{f}(v) \\ w \neq u}} Z_w^{s[u]+\gamma} \right) \left(\prod_{\substack{v = \tilde{f}(v) \\ v \neq u}} X_v^{s[u]+\gamma} \right). \end{aligned} \quad (\text{B.12})$$

After performing this commutation, there are no classically controlled operations to be inserted to the left of the shift operator which will depend on $s[u]$; we may then commute it past all additional operations and remove it from the expression. Then, rather than inserting shift operations and the corrections described, we may accumulate the corrections described on the right-hand side after the measurement on u . The amount of work required for each measurement in this alternative procedure is then dominated by the work required to update the corrections, which is $O(n \deg[\tilde{f}(u)])$; summed over all vertices u , obtaining a normal form may then be done in time $O(nm)$.

B.2.5 Total run-time complexity

Comparing the complexity of the phases above (and recalling the relationship between n and m with N , M , and k at the beginning of Section B.2.4, the run-time of both the

DKP and RBB constructions are then dominated by the phase of obtaining a 1WQC procedure in normal form. This requires time $O(nm)$ for both constructions, which evaluates to $O((k + N)M)$ for the DKP construction and $O((N + M)NM)$ for the RBB construction. This bound on the run-time coincides with that of the semantic map S described in Section 3.5.